
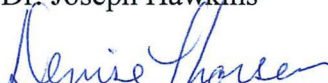


ATTITUDE DETERMINATION FOR SMALL SATELLITES USING GPS
SIGNAL-TO-NOISE RATIO

By
Daniel Peters

RECOMMENDED:



Dr. Joseph Hawkins

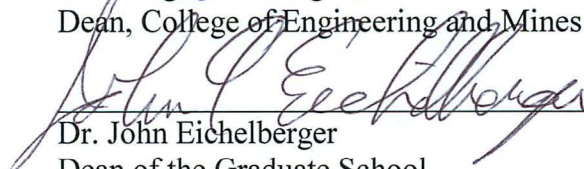

Dr. Denise Thorsen


Dr. Dejan Raskovic, Advisory Committee Chair


Dr. Charlie Mayer, Chair
Department of Electrical & Computer Engineering

APPROVED:


Dr. Douglas Goering
Dean, College of Engineering and Mines


Dr. John Eichelberger
Dean of the Graduate School


Date

ATTITUDE DETERMINATION FOR SMALL SATELLITES USING GPS
SIGNAL-TO-NOISE RATIO

A
THESIS

Presented to the Faculty
of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

By

Daniel Peters, B.S.

Fairbanks, Alaska

May 2014

Abstract

An embedded system for GPS-based attitude determination (AD) using signal-to-noise (SNR) measurements was developed for CubeSat applications. The design serves as an evaluation testbed for conducting ground based experiments using various computational methods and antenna types to determine the optimum AD accuracy. Raw GPS data is also stored to non-volatile memory for downloading and post analysis. Two low-power microcontrollers are used for processing and to display information on a graphic screen for real-time performance evaluations. A new parallel inter-processor communication protocol was developed that is faster and uses less power than existing standard protocols. A shorted annular patch (SAP) antenna was fabricated for the initial ground-based AD experiments with the testbed. Static AD estimations with RMS errors in the range of 2.5° to 4.8° were achieved over a range of off-zenith attitudes.

Table of Contents

	Page
Signature Page	i
Title Page	iii
Abstract	v
Table of Contents	vii
List of Figures	xi
List of Tables.....	xiii
List of Appendices	xv
Acknowledgements	xvii
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Motivation	2
1.3 Previous Works	2
Chapter 2: Attitude Determination Systems Overview	5
2.1 Methods of Attitude Determination	5
2.2 Inertial Measurement.....	5
2.3 Magnetometers	6
2.4 Star Trackers	6
2.5 Sun Sensors	6
2.6 Horizon Sensors	7
2.7 GPS Based ADS.....	7
2.7.1 GPS Carrier Wave Phase Difference.....	8
2.7.2 GPS SNR.....	9
Chapter 3: SNR ADS Theory and Design.....	11
3.1 GPS Overview.....	11
3.1.1 GPS Signals.....	11
3.2 SNR ADS	13
3.2.1 Vector Sum SNR ADS	13

3.2.2	Antenna Pattern SNR ADS Theory	13
3.3	Signal Strength Factors	17
3.3.1	SV Power Levels	18
3.3.2	Path Loss	19
3.3.3	SV Gain Patterns	22
3.3.4	Atmosphere	25
3.3.5	SV Yaw Attitude	26
3.3.6	Ionospheric Scintillation.....	27
3.3.7	Multipath	28
3.4	Antenna Characterization	29
3.5	GPS Receivers.....	29
3.5.1	Space Capability.....	29
3.5.2	Low Power	30
3.5.3	SNR Measurement Resolution	30
3.5.4	Update Rate	31
3.5.5	GPS Receiver Choices.....	31
Chapter 4:	Antenna Design	33
4.1	Conventional GPS Antennas	33
4.1.1	Conventional Microstrip Patch Antenna	34
4.1.2	Choke-Ring Patch Antenna	35
4.1.3	Pinwheel Antenna.....	36
4.1.4	Shorted Annular Patch Antenna	37
4.2	Shorted Annular Patch Antenna Theory.....	37
4.3	SAP Antenna Design.....	39
4.4	SAP Antenna Measurements	40
4.4.1	SAP Electrical Measurements	40
4.4.2	SAP Radiation Pattern Measurements.....	42
Chapter 5:	Testbed Design	45
5.1	Design Overview	45
5.2	Hardware Revision 1.2 Design.....	47
5.3	Hardware Revision 2.0 Design.....	48

5.3.1	MSP430 54xx Series Issues	48
5.4	Hardware Revision 2.1 Design.....	50
5.4.1	Power Management.....	50
5.4.2	Clock Generation.....	52
5.4.3	Memory Storage	52
5.5	IPC Protocol	53
5.5.1	PIPC Protocol Operation.....	53
5.5.2	PIPC Protocol Media Access	57
5.5.3	SPI Protocol Operation.....	59
5.5.4	Inter-Processor Communication Performance Measurements	61
5.5.5	Inter-Processor Communication Pin Requirements	65
5.5.6	Inter-Processor Communication Applications	65
5.6	Testbed Power Measurements.....	66
5.7	Testbed Hardware Summary	67
Chapter 6: Software Development		69
6.1	Software Development Introduction	69
6.2	Real-Time Operating System Overview	69
6.3	SYSBIOS Real-Time Operating System.....	70
6.4	Testbed Programs	71
6.4.1	Program Tasks.....	74
6.4.2	Program HWIs.....	75
6.4.3	Program SWIs	75
6.5	File Formats.....	75
6.6	Testbed Operation	77
6.6.1	GPS SV Skyplot	78
6.6.2	GPS SNR Levels	78
6.6.3	Attitude Display	78
6.6.4	DMA Transfers.....	78
6.6.5	Battery Status	78
6.7	Software Development Summary	79
Chapter 7: Test Setup and Results.....		81

7.1	Test Setup.....	81
7.2	Test Measurement Sessions.....	81
7.3	Attitude Calculations.....	82
7.3.1	Normalized SNR Calculation Set.....	84
7.3.2	Weighting Matrix Calculation Set.....	85
7.3.3	SV Radiation Pattern and Path Loss Correction Calculation Set	86
7.3.4	SV Output Power Correction Calculation Set	87
7.3.5	Atmospheric Loss Correction Calculation Set	87
7.4	Calculation Set Summary	88
7.5	Comparison to Previous Results.....	89
Chapter 8: Conclusion and Future Work.....		93
References		95
Appendices		99

List of Figures

Figure 1: GPS Carrier Phase Vector Relationship	8
Figure 2: Antenna Off-boresight Angle Relative to SV LOS Vector	14
Figure 3: Geometric Representation Example of Equation 4.....	15
Figure 4: SV Received Power Levels measured as SNR in AMU units.	19
Figure 5: Varying Signal Distances for Users on the Earth	20
Figure 6: Varying Signal Distances for a Satellite in LEO	21
Figure 7: Path Loss of GPS signals relative to $\theta=0$	22
Figure 8: GPS SV Radiation Pattern	22
Figure 9: SV Transmit Antenna Radiation Patterns	24
Figure 10: Atmospheric Attenuation at the Earth's Surface	26
Figure 11: SV Yaw Attitude Changes to Optimize Solar Panel Orientation in Orbit	27
Figure 12: Conventional Patch Antenna (a) Example of Theoretical and Actual Radiation Pattern Comparison, and (b) Miniaturized High Permittivity Dielectric Square Patch...	35
Figure 13: Choke-Ring Antenna.	35
Figure 14: Spiral Pinwheel Antenna Structure (Source: Kunysz, NovAtel Inc.)	36
Figure 15: Theoretical and Actual Radiation Pattern for a SAP Antenna.....	37
Figure 16: SAP Antenna Geometry: Top and Lateral Views.....	38
Figure 17: SAP Antenna	40
Figure 18: SAP Network Analyzer VSWR S11 Measurements	41
Figure 19: SAP Network Analyzer S11 Insertion Loss Measurements	41
Figure 20: SAP S11 Smith Chart Measurement.....	42
Figure 21: SNR-to- α Mapping: Measured AMU Samples (Blue) and Best Fit (Red)	43
Figure 22: SNR-to- α Mapping Best Fit from Figure 21 in dB-Hz Units	44
Figure 23: SAP Antenna Radiation Pattern from Best-Fit in Figure 22.....	44
Figure 24: Testbed Block Diagram	45
Figure 25: Testbed PCBs of (a) Revision 1.2, (b) Revision 2.0 and (c) Revision 2.1.....	47
Figure 26: Testbed PCB Revision 2.1 Block Diagram.....	50
Figure 27: PIPC Synchronization Signals	54
Figure 28: PIPC Protocol Trigger Timing Diagram.....	56
Figure 29: PIPC Protocol Data Transfer Cycles	57
Figure 30: PIPC Arbitration Signals	58

Figure 31: PIPC Arbitration Timing Diagram.....	59
Figure 32: SPIIPC Inter-Processor Signals	60
Figure 33: IPC Protocol Maximum Throughput	61
Figure 34: IPC Protocol Effective Throughput	62
Figure 35: IPC Protocol Current Consumption	63
Figure 36: IPC Protocol Current Ratio.....	64
Figure 37: IPC Protocol Switching Current Ratio.....	64
Figure 38: Testbed MCU0 Program Data Flow	72
Figure 39: Testbed MCU1 Program Data Flow	73
Figure 40: GPS File Formatting	76
Figure 41: Testbed Menu	77
Figure 42: Measurement Test Apparatus Setup	81

List of Tables

Table 1: SV Output Power	19
Table 2: SV Block Members	23
Table 3: SV Antenna Radiation Pattern Coefficients (dB)	24
Table 4: GPS Satellite Constellation	25
Table 5: GPS Receiver Features.....	31
Table 6: SAP Design Parameters	40
Table 7: SAP Antenna Mapping Function Coefficients.....	43
Table 8: Testbed Hardware Revision Differences.....	47
Table 9: IPC Protocol Pin Requirements	65
Table 10: Testbed Current Consumption	66
Table 11: GPS File Data Types	76
Table 12: Measurement Sessions	82
Table 13: Attitude Estimation Results based on Normalized SNR Calculation Set	84
Table 14: Attitude Estimation Results based on Weighting Matrix Calculation Set	85
Table 15: Attitude Estimation Results based on SV Radiation Pattern and Path Loss Correction Calculation Set.....	86
Table 16: Attitude Estimation Results based on SV Output Power Correction Calculation Set	87
Table 17: Attitude Estimation Results based on Atmospheric Loss Correction Calculation Set	88
Table 18: Mean Estimation Results for Calculation Sets	88
Table 19: Zenith Pointing Attitude Estimation Results for Ground Based Measurements	89
Table 20: Attitude Estimation Results for TANS Hardware Measurements by Behre	91

List of Appendices

Appendix A: Testbed Hardware Schematics.....	99
A.1 Testbed Version 2.1 Power Management Circuits	99
A.2 Testbed Version 2.1 MCU0 Circuits	100
A.3 Testbed Version 2.1 MCU1 Circuits	101
Appendix B: Attitude Determination MATLAB Code	103
B.1 MATLAB Constant Declarations	103
B.2 MATLAB Script for Loading Testbed GPS Log File	105
B.3 MATLAB Script for Loading PolaRX Reference GPS Log File	106
B.4 MATLAB Script for Calculating Attitude and Presenting Statistics.....	107
B.5 MATLAB Script for Generating User Interface Program	111

Acknowledgements

I would like to thank UAF, the Geophysical Institute, and Alaska Space Grant Program for granting me research assistantships during my graduate studies. I am grateful to Dr. Raskovic, Dr. Thorsen and Dr. Hawkins for their teaching, support, and patience during my academic career. To my wife Anjee, for contributing to this thesis by giving much more than your share of raising our boys while being stretched thin so I could finish, thank you!

Chapter 1: Introduction

1.1 Overview

A relatively new class of small satellites is emerging for conducting orbital research and science. The smallest of these are the Pico satellites, which have a mass less than 1 kg, and also Nano satellites which have a mass less than 10 kg. A standardized type of Pico satellite is referred to as a cube satellite, or CubeSat, measuring only 10 cm per side for a one unit (1U) cube. Sometimes multiple CubeSats are joined to make a small Nano satellite. For example, a Nano satellite consisting of three cubes would be designated as a three unit (3U) CubeSat. The modest dimensions and standardized deployment method from the launch vehicle make the design an extremely cost-effective way to conduct orbital science, often with a budget of under \$100,000. Emerging but unproven technologies can be implemented with minimal risk.

The challenge for designing small satellites is to keep the hardware size, mass and power consumption to an absolute minimum while simultaneously keeping the cost down. Commercial-off-the-shelf (COTS) products are used in place of specialty hardware designed by costly aerospace contractors. A lot of development is happening in this rapidly growing niche, especially at universities where student teams often design and build the satellites.

This thesis details the hardware development and testing of an attitude determination system (ADS) specifically designed for small satellite applications using GPS signal-to-noise ratio (SNR). Compared to other ADSs commonly used, the SNR ADS is well suited for small satellite applications because of its low complexity and robustness. However, a limitation of SNR ADS designs is relatively low accuracy. The design presented in this thesis aims to improve upon the accuracy performance previously attained. A low power, dual-processor testbed platform using a real time operating system (RTOS) is developed which allows various types of GPS receivers, algorithms, and antenna types to be easily evaluated for accuracy

performance. A custom antenna is designed for optimal accuracy, and the results are compared to previous works.

1.2 Motivation

Magnetometers combined with Sun sensors are currently used for some CubeSat designs [1], [2], mainly because of their low cost, simplicity, small size and low power consumption. A GPS system may also be used to provide the positional information required for the magnetometer attitude solution. If a small satellite is already equipped with a GPS, then a SNR ADS can be adapted with very little overhead in terms of space and power consumption. The GPS ADS in this scenario negates the need for a Sun sensor when a magnetometer is also used. A GPS / magnetometer ADS can provide a 3-axis attitude solution with only one GPS antenna. A two antenna SNR ADS could also be used to obtain a 3-axis solution, but in this case either a GPS receiver with multiple antenna inputs or two GPS receivers are required.

1.3 Previous Works

At the time of this writing, the most accurate non-simulated single antenna results have been obtained by the GPS Attitude/Navigation Experiment (GANE) in [3], where root mean square (RMS) errors of 3.2° were achieved in orbit using a choke-ring patch antenna pointing in the zenith direction. The same experiment had RMS errors of 5.2° when the antenna was pointed 45° off-zenith. In another study with the Federation Satellite (FedSat), attitude estimations were largely affected by a non-optimal antenna direction and distortions in the radiation pattern caused by an offset ground plane [4]. A conventional patch antenna was used, yielding an accuracy limited to 14° RMS error.

Hardware-in-the-loop orbital simulations in [5] have demonstrated better results. RMS errors of 2.5° were achieved using a software-defined antenna radiation pattern and Kalman filtering. Using a software-defined antenna radiation pattern is highly beneficial for the simulation, because the non-ideal characteristics of a real

antenna are removed. The simulated radiation pattern was modeled with a maximum signal strength along the boresight vector and a zero signal strength below the local horizon of the antenna. The modeled antenna radiation pattern is optimum, such that there is a maximum distinguishable off-boresight angle per varying SNR.

Some distinctions should be made about the results of the comparison studies in [3-5]. The experiments and simulations were performed under orbital conditions, since the intended application is for satellites. The dynamics are much different than ground based environments. The higher carrier Doppler variation experienced in an orbital environment will affect how the GPS tracking loops operate, and may subsequently degrade the accuracy compared to ground measurements where there is less Doppler shift. On the other hand, the space environment has less potential for multipath and a higher potential for GPS satellite visibility. This should improve the accuracy compared to ground measurements where obstacles and multipath are prevalent. Some areas of the Earth have better GPS satellite vehicle (SV) coverage than others and the results will vary somewhat by the geographical location due to the GPS constellation. Conversely, an orbiting spacecraft will have a more diverse SV view during the course of the orbit. Even with all the differences between environments, comparisons in [6] between ground measurements and orbital simulations revealed similar results for orbital and ground accuracy performance. In another study of the GANE experiment [7], zenith pointing antenna measurements revealed orbital performance was about 30% better compared to ground measurements using the same kind of antenna. The best ground based performance was achieved in [7] with 4.2° RMS error when a zenith pointing choke-ring antenna was used.

This thesis aims to develop a GPS SNR based ADS specifically for use on a CubeSat. The design goals of the hardware are primarily about developing a working prototype that can be evaluated for its performance to determine if the design should be developed further for a future CubeSat mission.

In Chapter 2 there is a brief overview of seven methods commonly used for ADS on satellite systems including magnetometers, inertial sensors, star trackers, Sun

sensors, horizon sensors, GPS carrier wave, and GPS SNR. The merits of each type of ADS pertaining to CubeSats are summarized.

GPS related ADSs using SNR are described in more detail in Chapter 3 where algorithms are presented for acquiring attitude estimations based on GPS data. Factors affecting the SNR are studied to isolate the elements not related to the antenna attitude. Several GPS receiver module candidates are evaluated for possible use in a space environment.

Chapter 4 explores the different types of antennas used in GPS systems. The strengths and weaknesses of each are evaluated as they pertain to GPS SNR attitude determination. A specialized shorted annular patch (SAP) antenna is examined and measurements of a fabricated prototype antenna are presented.

Chapter 5 contains the hardware details of a dual processor embedded testbed that was designed and constructed. Design iterations during the development of the testbed are presented. A newly developed inter-processor communication protocol is explained and evaluated for transfer performance and power consumption.

Software development on the testbed hardware is discussed in Chapter 6. A summary of real time operating systems is presented with an overview of the testbed software implementation. The user operation of the testbed is also explained.

In Chapter 7, the GPS measurements collected by the testbed hardware are presented with analysis of the attitude performance accuracy. Data from 5 static attitude measurement sessions is assessed where elements of SNR signal influences are incrementally included in calculations to observe the effect on accuracy. The attitude estimation results are compared to previous works of GPS SNR based ADS.

Chapter 2: Attitude Determination Systems Overview

2.1 Methods of Attitude Determination

Seven commonly used ADS methods [1], [2], [6], [7] are compared in this chapter. The methods examined are used on satellites in general, including those costing millions of dollars and having mass in excess of 1000 kg. Such large satellites have systems and redundancies that emphasize accuracy and reliability. In contrast, the design criterion for a CubeSat is very restrictive in terms of complexity, size, cost, and power consumption. Each type of ADS is examined in terms of the suitability for use on a CubeSat.

2.2 Inertial Measurement

One simple and traditional approach to attitude determination (AD) is to use an inertial measurement unit (IMU) which works by measuring the outputs of accelerometers and gyroscopes. Advances in micro electro-mechanical systems (MEMS) technology has allowed inexpensive IMU sensors to become widely available. MEMS sensors are miniature and low power compared to their mechanical counterparts, thus making their use ideal in small low cost satellites. IMU sensors directly measure the derivatives of attitude and position, with the solution being computed by integrating the output of the sensors. The attitude solution will begin to drift relative to the actual attitude and accrue significant error over a period of days, since even the smallest bias errors in the rate measurements accumulate over time with integration. In the short term, IMU sensors easily achieve a sub-degree level of accuracy based on accurate initial conditions.

An IMU must be used in conjunction with another form of attitude determination for two reasons: the initial conditions are usually not known, and the accumulated error must be corrected. For CubeSat deployments, the initial attitude is often not known and difficult to accurately predict.

2.3 Magnetometers

COTS magnetometers have about 1° accuracy if the field vector is accurately known. By measuring the Earth's magnetosphere using magnetometers, satellites in orbit are able to measure the field orientation and deduce their attitude. However, the Earth's magnetic field is not uniform and has varying characteristics depending on the geographic position. For instance, the field is mostly parallel to the Earth's surface near the equator, but the field is nearly perpendicular to the Earth's surface near the poles. These variances make it necessary to know the satellite's position within the magnetic field, and compare the readings to a field model before making an attitude calculation. Magnetic interference caused by satellite electronics and unpredictable ionospheric currents, and on-board satellite electronics contribute to errors of the measurements. Since the orbit of the satellite is generally known *a-priori*, the ADS can use an estimated position and compute the attitude based on a field model. Alternatively, a GPS can provide the positional information for the calculation. The magnetometer-based ADS is currently one of the preferred methods for CubeSat applications.

2.4 Star Trackers

Star trackers use optical sensors to compare the known location of star constellations with an onboard database. Significant processing and power consumption is required to obtain an attitude solution. Interference from other light sources such as the Sun, Moon and Earth can inhibit the star sensor from computing an attitude solution. Although this imaging technique can be a very accurate method of attitude determination ($<0.1^\circ$), the hardware is on the same order of magnitude in mass as a cube satellite, and not suitable for CubeSat applications.

2.5 Sun Sensors

Simple sensors or even solar panels can be used to obtain a vector angle from the sensor to the Sun. Only a single reference is obtainable, as the sensor cannot distinguish any rotation along the axis of the vector, and so a full 3-axis solution is not

possible. Sun sensors are commonly used on CubeSat designs, but a major disadvantage to Sun sensors is that when the satellite falls within the Earth's shadow during a portion of its orbit, no attitude information can be measured. Sun sensors can provide accuracies ranging from several degrees to less than 0.1 degrees.

2.6 Horizon Sensors

Horizon sensors operate on the same premise as Sun sensors and have similar limitations in terms of outages while in the Earth's shadow. The Earth Horizon is the reference that replaces the Sun vector. Accuracy is generally less than Sun sensors because of the atmosphere creates a gradient that is poorly defined along the horizon resulting in accuracies of about 1.0 degree.

2.7 GPS Based ADS

While GPS was initially intended to provide position and velocity information, other uses have been developed, such as precise timing and AD. GPS based AD can be accomplished in two ways; Carrier Wave Interferometry or Signal-to-Noise Ratio (SNR). Both methods have some limitations when used in the orbital environment: All COTS GPS modules have software limits so that the device will not operate if it exceeds speed and altitude thresholds of 515 m/s and 18,300 m – parameters that are far surpassed by satellites. The limitation was imposed by the Coordinating Committee for Multilateral Export Controls (COCOM) for security reasons [8]. This presents a major problem for budget sensitive missions attempting to use COTS components. To overcome the problem, custom GPS hardware not subject to COCOM constraints has been used in a small satellite mission [9]. Also, some COTS GPS modules have been found to still provide useful raw data information, even when the COCOM limits are exceeded [10].

Even if a COTS GPS module is useable beyond the COCOM limits, the operating environment of an orbiting satellite is much different than the typical land-based use of a GPS module. The GPS frequency tracking loops must be able to cope with the higher carrier wave Doppler shifts encountered due to the extreme velocity of

the satellite. The antenna may not be able to receive signals from enough GPS SVs if the attitude is unfavorable. Such an attitude might be when the antenna is pointed towards the horizon, or in the worst case, directly at the Earth.

The major advantage of using a GPS-based ADS arises when a GPS is already being used in the payload for positioning or precise timing purposes. If this is the case, there is only a small computational overhead to implement the ADS, especially for SNR ADS.

2.7.1 GPS Carrier Wave Phase Difference

If three or more GPS antennas are used and the carrier phase of GPS signals are measured simultaneously, the baseline vectors between antennas can be determined and the attitude orientation defined by these vectors can be calculated. Figure 1 depicts the relationship between the incident carrier wave \hat{e} vector originating from a SV, the baseline vector of two antennas \mathbf{r} , and the difference of phase ϕ of the signal received between two antennas. A third antenna may be mounted orthogonally to the first two to compute a 3-axis solution.

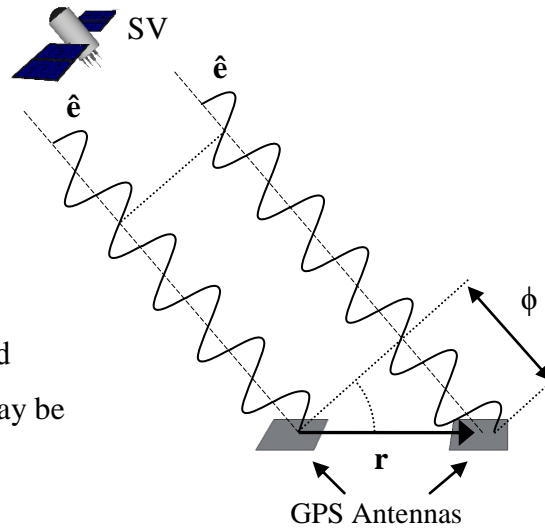


Figure 1: GPS Carrier Phase Vector Relationship

Sub-degree accuracy is possible for baselines of about 1 m. Although carrier phase achieves precise sub degree accuracy, there are some drawbacks when considering this method for implementation in a CubeSat. The accuracy is proportional to the antenna baseline length, and thus the GPS antennas need to be separated by relatively large distances. This presents a problem on a CubeSat because it requires the extension of antenna booms once the satellite has been deployed from the launch vehicle. Precisely deploying such booms from a small satellite adds

complexity and a certain risk of deployment failure. Alternatively, the three antennas could be separated by a very short baseline, but the RMS accuracy would be degraded to about 5° for a sub-10 cm baseline [11]. Complicating the task is that a specialized GPS receiver capable of multiple antenna input channels is needed to make the precise phase measurements. There are certain GPS receivers designed for AD such as the Trimble Tans Vector, which is capable of processing signals from multiple antennas simultaneously. Unfortunately, excessive mass and power consumption prohibit their use for CubeSats.

2.7.2 GPS SNR

GPS signal strength is highly correlated to the angle at which the signals arrive to the antenna according to the receiving antenna gain pattern. Other factors also affect the signal strength and will be discussed in Chapter 3. Higher signal strengths will be observed when the antenna is pointing directly towards the transmitting GPS satellite vehicle, and decreasing as the angle is increased. This relationship is a function defined by the particular antenna radiation pattern. The line-of-sight (LOS) vector for each tracked satellite, along with signal strength in terms of SNR is available from the GPS receiver module output data packet. The measurement is usually given normalized to a 1 Hz bandwidth as carrier-to-noise-zero (C/N_0) with the units of dB-Hz. Some GPS receivers also use Amplitude Measurement Units (AMU) for the C/N_0 measurements. The C/N_0 ratio and SNR will be considered interchangeable for the purpose of this study. The SNR and vector information can be used to calculate an attitude vector of the receiver antenna if the receiver antenna radiation pattern is known. If a single antenna is used, only a 2-axis attitude solution is attainable, but a full 3-axis solution is possible when two or more non-aligned antennas are used. The disadvantage of using multiple receivers is higher power consumption and payload occupancy. A single GPS receiver with multiple antenna inputs would be ideal, but is much more expensive and not readily available as a COTS component. Previous studies using single-antenna SNR GPS ADS's have

shown accuracy errors in the range of 3° RMS in orbit and about 4° RMS on the ground [7].

Chapter 3: SNR ADS Theory and Design

3.1 GPS Overview

The Global Positioning System, known ubiquitously as GPS, was developed by the U.S. Department of Defense for military purposes and was later made available for the general public [12], [13]. Its main purpose is to allow users to obtain their coordinates anywhere on Earth using a GPS receiver. Many other uses have since been developed, such as time synchronization, surveying and attitude determination.

The principle behind the operation of the GPS is that a GPS receiver is able to calculate its own position and velocity using radio signals that are simultaneously received from multiple GPS SVs. The GPS constellation consists of at least 24 operational SVs but may contain up to 32 SVs at any given time. Each SV has an atomic clock capable of keeping an extremely accurate and stable time reference. The SV sends a continuous radio signal encoded with many parameters including the precise time and orbit parameters of the SV. The GPS receiver requires signals from at least 4 SVs to calculate its 3D position on the Earth in terms of longitude, latitude and altitude. The position is then calculated by trilateration, where the elapsed time for a signal to reach the receiver from a SV at the speed of light is equated to a distance. Normally only 3 SV signals would be required to obtain a 3D solution, but a forth signal is needed because the GPS receiver clock is not precise— thus the clock error of the receiver becomes the forth variable in the system of equations. The inner workings and the algorithms used by the GPS receiver to perform this process is quite complex and not within the scope of this thesis.

3.1.1 GPS Signals

The GPS SVs broadcast data towards the Earth on the frequencies 1575.42 MHz and 1227.60 MHz, named L1 and L2 respectively. Data is transmitted using binary phase shift key (BPSK) as the carrier wave modulation technique and code division multiple access (CDMA) to distinguish the SV signals using the same frequency. L1 carries both the civilian coarse-acquisition (C/A) code and the military

encrypted precision (P) code. L2 carries only the P code. COTS GPS modules are generally limited to using the C/A code on the L1 frequency.

The C/A code consists of 32 different pseudorandom numbers (PRN) at a rate of 1.023 million chips per second, and the P code PRN is 10.23 million chips per second. The C/A PRNs repeat once per millisecond, while the P PRNs are much longer and repeat once per week.

Since all the GPS SVs transmit on the same frequency, the GPS receiver distinguishes one signal from another by correlating the PRN of the received signal with its own copy of the PRNs. Once the receiver has locked onto a signal by phase shifting its own PRN sequence to match the incoming signal, it is able to decode the GPS information. The information is modulo-2 added at 50 bits/s onto the PRN, such that the low bit rate of the information will have little effect on the overall correlation of the much higher chips per second PRN. Each operational SV is assigned one of the 32 PRNs, but these assignments are not fixed. New SVs are launched to replace aging SVs, and often the GPS constellation consists of spare SVs that are only activated with an assigned a PRN when another operating SV fails.

The information transmitted by each SV consists of two primary segments: the ephemeris and the almanac. The ephemeris contains information only about the SV that the signal was received from, so each SV transmits a unique ephemeris. It repeats every 30 seconds, and contains precise orbit parameters of the SV. The almanac contains coarse orbit information about all operating SVs, with all SVs transmitting the same segment. Ionospheric parameters are also included, requiring 12.5 minutes to complete the transmission segment.

The signal levels from the GPS SV on or near the Earth are very weak— as low as -160 dBW [14]. This is well below the thermal noise floor and would otherwise not be detectable if it were not for the processing gain of about 50 dB that is achieved by correlation of the PRN sequences. The low signal level is because of significant path losses from approximately 20,000 km of propagation. Path loss is discussed in more detail later in the chapter.

3.2 SNR ADS

There are two ways to estimate a two-axis attitude vector using SNR measurements from a single antenna. The first method involves a simple vector sum calculation, and requires only a SNR measurement from one SV to obtain the attitude estimation. The second method equates the measured SNR of each received signal to an off-boresight angle based on the receiving antenna radiation pattern.

3.2.1 Vector Sum SNR ADS

The Vector Sum attitude estimation is calculated by summing the visible SV LOS vectors, each with a magnitude equal to the SNR value. The solution is simply

$$\bar{\mathbf{B}} = \sum_{i=1}^n \text{SNR}_i \hat{\mathbf{L}}_i, \quad (1)$$

where \mathbf{B} is the antenna boresight vector, \mathbf{L} is the LOS unit vector to the i^{th} GPS SV, SNR is the signal to noise ratio of the i^{th} GPS SV signal, and n is the number of SV signals visible to the GPS receiver. The accuracy is generally low, on the order of 15° [5], and is highly dependent on the SV constellation geometry. This method will not be further explored because of its inferior performance compared to the Antenna Pattern method.

3.2.2 Antenna Pattern SNR ADS Theory

The second method relies on the dot product property

$$\cos\alpha = \bar{\mathbf{B}} \cdot \bar{\mathbf{L}} = B_x L_x + B_y L_y + B_z L_z, \quad (2)$$

where \mathbf{B} is the antenna boresight unit vector and \mathbf{L} is the LOS unit vector to the GPS SV referenced to a coordinate frame, and α is the angle between \mathbf{B} and \mathbf{L} [5]. Figure 2 depicts a zenith pointing antenna boresight vector \mathbf{B} and LOS vector \mathbf{L} for a particular SV. The boresight vector \mathbf{B} is defined as the attitude vector of the antenna, and when the antenna is attached to a CubeSat it becomes the unknown attitude vector we are interested in determining. \mathbf{L} is an output vector provided by the GPS receiver for each SV from which it is receiving data.

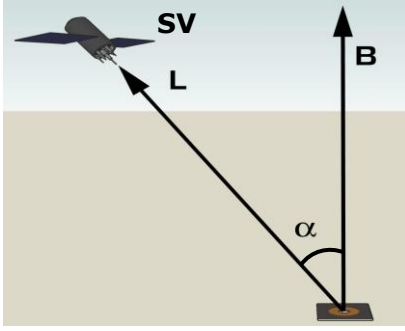


Figure 2: Antenna Off-boresight Angle Relative to SV LOS Vector

In order to solve for the boresight (attitude) vector \mathbf{B} , both α and \mathbf{L} are required, but α cannot be obtained directly as it is not a value that is provided by GPS receivers. However, GPS receivers do provide a SNR measurement for each SV as an output that can be correlated to the angle α . The relationship between the SNR and α depends largely on the radiation pattern of the GPS receiver antenna. A SNR-to- α conversion mapping is

performed for the GPS receiver antenna to get the desired angle α :

$$\alpha = f(\text{SNR}). \quad (3)$$

The SNR-to- α conversion is not perfect because of the inherently noisy and fluctuating SNR readings, but despite these fluctuations, an attitude vector estimation is possible from the available data. When \mathbf{L} and α measurements from three separate SVs are available, the linear equation to solve for the attitude is

$$\begin{aligned} \mathbf{A}\mathbf{B} &= \mathbf{S} \\ \begin{bmatrix} L_1\hat{x} & L_1\hat{y} & L_1\hat{z} \\ L_2\hat{x} & L_2\hat{y} & L_2\hat{z} \\ L_3\hat{x} & L_3\hat{y} & L_3\hat{z} \end{bmatrix} \begin{bmatrix} B\hat{x} \\ B\hat{y} \\ B\hat{z} \end{bmatrix} &= \begin{bmatrix} \cos\alpha_1 \\ \cos\alpha_2 \\ \cos\alpha_3 \end{bmatrix}, \end{aligned} \quad (4)$$

where \mathbf{A} is the coefficient matrix of the \mathbf{L} vectors, \mathbf{B} is the unknown attitude vector, \mathbf{S} is the $\cos(\alpha)$ measurements derived from the receiving antenna SNR-to- α mapping function, and the subscripts denote the index of the SV. \mathbf{L} in Equation 4 is expressed in Cartesian coordinates, though the vector is provided by the GPS receiver in Polar coordinates with Elevation and Azimuth components. The conversion of the unit vector \mathbf{L} from Polar to Cartesian is

$$\begin{aligned} x &= \sin\phi \cos\beta \\ y &= \sin\phi \sin\beta, \\ z &= \cos\phi \end{aligned} \quad (5)$$

where ϕ is the SV elevation and β is the SV azimuth, the positive \hat{x} direction points North, and the positive \hat{y} direction points West, and the positive \hat{z} direction points up.

Equation 4 can be visualized in Figure 3 where there are three SV LOS \mathbf{L} vectors represented by colors blue, green and red. Each LOS vector has an associated plane, normal to the vector. The distance of each plane from the origin is equal to $\cos(\alpha)$. The solution of attitude \mathbf{B} points to the location where all three planes intersect.

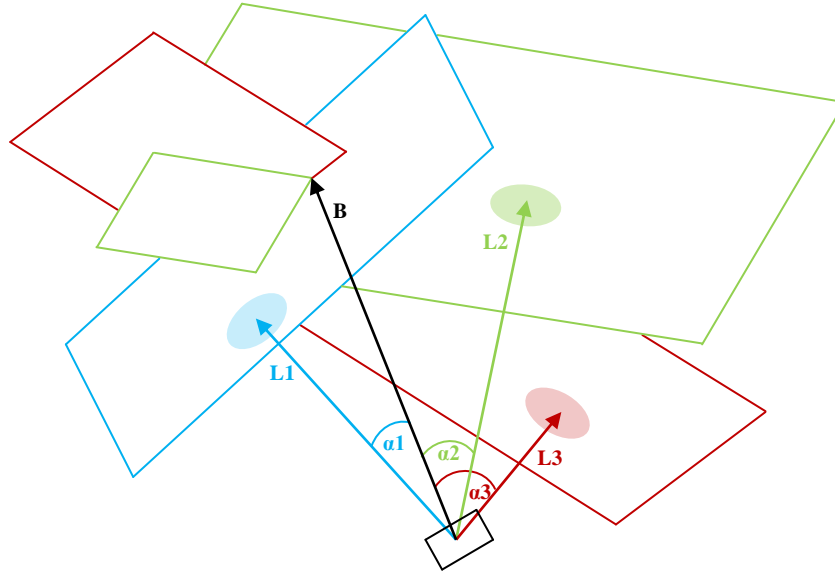


Figure 3: Geometric Representation Example of Equation 4

When there are more than three SV signals available, the over-determined system of equations will not have an exact solution because of the inherently noisy SNR data. For example, a fourth vector might yield three solutions of three intersecting planes, but no solution where four planes exactly intersect. For cases when the number of available SV signals is greater than three, a best fit solution, i.e., a solution that is the closest the algebraic least squares equation [12] is used

$$\mathbf{A}^T \mathbf{A} \bar{\mathbf{B}} = \mathbf{A}^T \bar{\mathbf{S}}$$

$$\begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} L_1 \hat{x} & L_1 \hat{y} & L_1 \hat{z} \\ L_2 \hat{x} & L_2 \hat{y} & L_2 \hat{z} \\ \vdots & \vdots & \vdots \\ L_n \hat{x} & L_n \hat{y} & L_n \hat{z} \end{bmatrix} \begin{bmatrix} B \hat{x} \\ B \hat{y} \\ B \hat{z} \end{bmatrix} = \begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_n \end{bmatrix}, \quad (6)$$

where n denotes the number of available SV signals.

A weighting factor of each SV signal can be applied to reflect the general quality of measurement based on certain parameters. Expanding Equation 6 to include the weighted least squares solution [7], [12] gives

$$\begin{aligned} \mathbf{A}^T \mathbf{W} \mathbf{A} \bar{\mathbf{B}} &= \mathbf{A}^T \mathbf{W} \bar{\mathbf{S}} \\ \begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ 0 & \cdots & w_n \end{bmatrix} \begin{bmatrix} L_1 \hat{x} & L_1 \hat{y} & L_1 \hat{z} \\ L_2 \hat{x} & L_2 \hat{y} & L_2 \hat{z} \\ \vdots & \vdots & \vdots \\ L_n \hat{x} & L_n \hat{y} & L_n \hat{z} \end{bmatrix} \begin{bmatrix} B \hat{x} \\ B \hat{y} \\ B \hat{z} \end{bmatrix} &= \end{aligned} \quad (7)$$

$$\begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ 0 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_n \end{bmatrix},$$

where \mathbf{W} is a weighting matrix of the measurements. Each W_n is optimized so that signals that are more reliable are more heavily weighted. For most antennas, signals near the boresight are more consistent, while signals towards 90° off-boresight experience more edge diffraction and have more variance due to side lobes in the radiation pattern. In addition, SVs with higher elevation referenced to Earth are less susceptible to multipath signals. From these empirical observations, the weighting matrix is set to:

$$W_n = \begin{cases} \cos^2(\alpha_n) + \sin^2(\phi_n) & \text{for } \alpha \geq \alpha^t \\ 0 & \text{for } \alpha < \alpha^t \end{cases}, \quad (8)$$

where n represents the SV PRN number, α is the off-boresight angle of SV LOS vector \mathbf{L} , α^t is a threshold angle and ϕ is the SV elevation. The α angle must be less than some set threshold so that only signals towards the front of the antenna are included in the calculation. Signals originating from behind the receive antenna are more likely to arrive at an angle where the actual radiation pattern has sharp lobes that diverge from the radiation pattern mapping function. For the SAP antenna, the threshold angle is set to 90° so signals arriving from the rear of the antenna are excluded from the calculation by setting the weight to zero.

3.3 Signal Strength Factors

Equation 3 relates the receiving antenna off-boresight angle solely on the receiving antenna radiation pattern characteristics. This is only a coarse first approximation, as there are other factors that affect to the received signal strength. SV transmitting output power, SV transmitting antenna gain, path loss and attenuation losses also affect the signal strength received from each SV. The general transmission equation [13] when expressed in dB is

$$Pr = Pt + Gt_{\theta} + Gr_{\alpha} + Lp_{\theta} + La_{\phi}, \quad (9)$$

where Pr is the received power, Pt is the transmitter output power, Gt is the transmitting antenna gain, Gr is the receiving antenna gain, Lp is the path loss and La is attenuation loss. The α subscript refers to the term dependence on the incident angle of the signal to the receive antenna, θ refers to the dependence of the signal angle to the transmitting SV antenna, and ϕ refers to the dependence on the elevation angle of the receive signal. Rearranging Equation 9 results in

$$Gr_{\alpha} = Pr - Pt - Gt_{\theta} - Lp_{\theta} - La_{\phi}. \quad (10)$$

Assuming the receiving antenna has a radiation pattern that is monotonic, i.e., there is a one-to-one relationship between the gain and the off-boresight angle α , the radiation pattern function can be inverted to express α in terms of the other variables

$$\alpha_{(Pr, Pt, \theta, \phi)} = f(Pr - Pt - Gt_{\theta} - Lp_{\theta} - La_{\phi}). \quad (11)$$

Using a GPS receiver, the received power Pr is not measured directly as an absolute value. Instead, the SNR is reported. Pr expressed in terms of SNR gives

$$\alpha_{(SNR, Pt, \theta, \phi)} = f(SNR - Pt - Gt_{\theta} - Lp_{\theta} - La_{\phi}). \quad (12)$$

When the SNR is measured in linear units as it is with the Trimble Copernicus GPS receiver module, Equation 12 is expressed as

$$\alpha_{(SNR, Pt, \theta, \phi)} = f\left(\frac{SNR}{Pt \cdot Gt_{\theta} \cdot Lp_{\theta} \cdot La_{\phi}}\right), \quad (13)$$

where SNR is in linear AMU units and the remaining variables are multiplication factors. The α in Equation 13 used in Equation 6 or 7 accounts for all the factors that

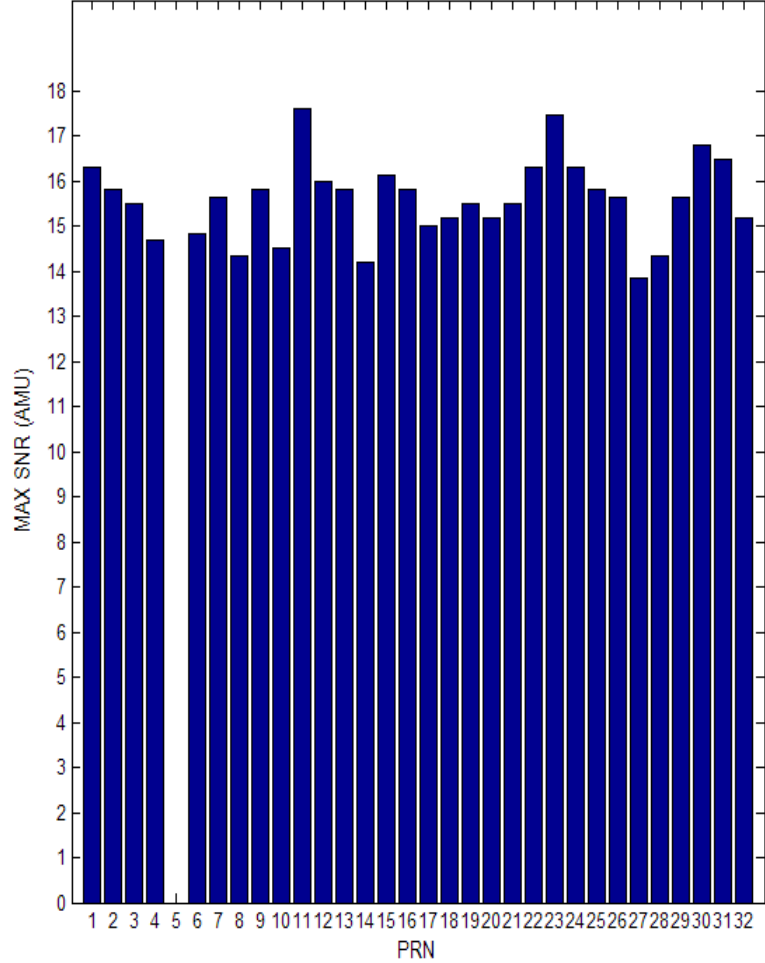
will be included in the attitude determination calculations later in Chapter 7. In [5], Lightsey assumes the effects other than SNR are smaller in magnitude than the noise seen in the SNR measurement, with no significant improvement to be expected from incorporating them into the measurement model. This assumption will be tested in Chapter 7 with the acquired measurement data. There are other factors affecting the signal strength including SV yaw attitude, scintillation and signal multipath which are difficult to model in real time and thus considered as noise. These factors will be discussed later in the chapter, but will be ignored in the calculations.

3.3.1 SV Power Levels

Each of the GPS SVs have slightly different output power levels. Figure 4 shows the measured power levels in terms of SNR of the SVs when the signal was at its maximum. This occurred with the receiving antenna pointing directly at the SV so that $\alpha = 0$, with corrections applied for space loss and transmitter radiation pattern. The measurements are recorded from the test sessions in Chapter 7. Table 1 shows the SV P_t values in terms of dB relative to the power output of the SV using PRN 1. It is important to note that if a PRN assignment changes, the new SV that transmits on the PRN code will almost certainly have a different transmission power level than the SV that was previously using the PRN code. This would necessitate an update to the referenced power levels that are shown in Table 1. No SV was assigned to PRN 5 at the time of the measurements as it was not an active code.

Table 1: SV Output Power

PRN	Relative Power Pt (dB _{PRN1})
1	0.00
2	-0.26
3	-0.45
4	-0.92
5	Not Available
6	-0.82
7	-0.35
8	-1.11
9	-0.26
10	-1.01
11	0.67
12	-0.18
13	-0.26
14	-1.21
15	-0.09
16	-0.26
17	-0.72
18	-0.63
19	-0.45
20	-0.63
21	-0.45
22	0.00
23	0.59
24	0.00
25	-0.26
26	-0.92
27	-1.41
28	-1.11
29	-0.35
30	0.26
31	0.09
32	-0.63

**Figure 4:** SV Received Power Levels measured as SNR in AMU units.

3.3.2 Path Loss

Electromagnetic signals originating from a point and propagating outwards will have a reduced power spatial density with increasing distance according to

$$\Phi = \frac{P_{rad}}{4\pi r^2} \quad , \quad (14)$$

where Φ is the power flux density, P_{rad} is the radiated power, and r is the distance from the source. Figure 5 illustrates how varying distances pertain to GPS users on Earth. From the perspective of the SV, there is a 13.88° span from the boresight to the

limb of the Earth based on a nominal GPS SV orbit radius of 26,560 km. The SV off-boresight angle is represented by θ . For a user on the Earth, the SV elevation represented by ϕ , ranges from 0° if the SV is at the horizon, and 90° if the SV is directly overhead. The SV is significantly closer if it is directly overhead compared to being at the horizon.

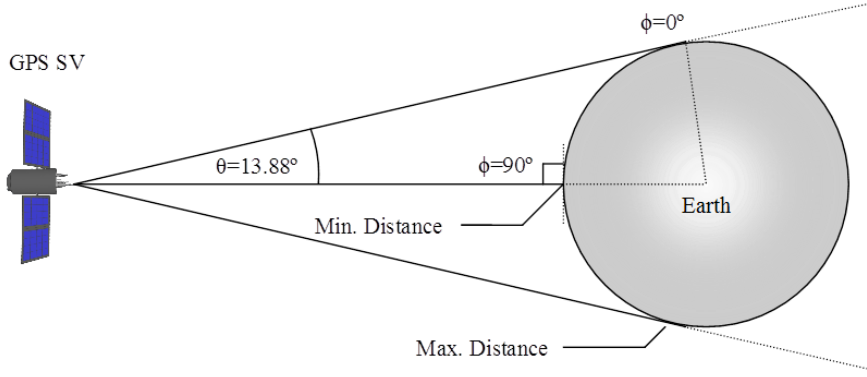


Figure 5: Varying Signal Distances for Users on the Earth

The distance from the SV to a user can be expressed as a function of the SV off-boresight angle

$$r = S_r \cos \theta - \sqrt{(E_r + U_a)^2 - S_r^2 \sin^2 \theta} , \quad (15)$$

where r is the distance from the SV to the user, S_r is the SV orbit radius, θ is the SV off-boresight angle, E_r is the Earth radius, and U_a is the user altitude. Equation 15 only applies when the LOS vector from the user to the SV is above the horizon and is otherwise undefined. Figure 6 depicts how the ratio of the maximum to minimum distances from the user to a SV slightly increases when the GPS receiver is in LEO.

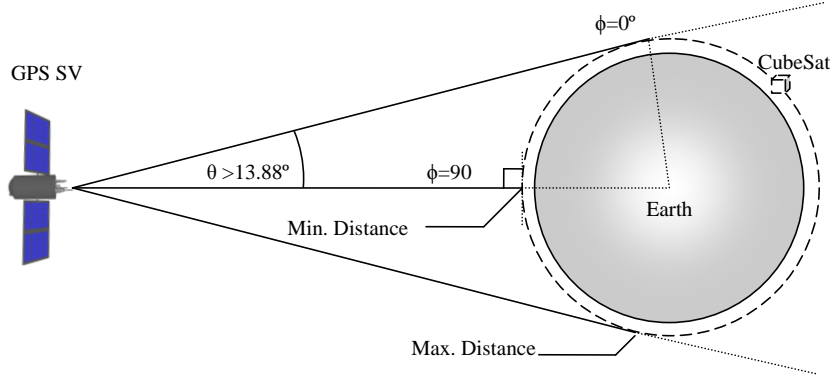


Figure 6: Varying Signal Distances for a Satellite in LEO

The path loss in Equation 15 is expressed in terms of the SV off-boresight angle θ . However, from the perspective of the GPS user, the SV elevation ϕ is measured as a component of \mathbf{L} , referenced to the horizon. The conversion of ϕ to θ is

$$\theta = \sin^{-1} \left(\frac{\cos \phi (E_r + U_a)}{S_r} \right), \quad (16)$$

where E_r is the Earth radius, S_r is the SV orbit radius, and U_a is the user altitude in km. For example, a CubeSat orbiting at 200 km would result in $\theta = 14.32^\circ$. With θ from (16) substituted into (15), and using r from (15) in (14) the path loss is calculated. The absolute value of Φ in (14) is not of interest, but rather the ratio of flux density variance over the range of angles θ . The most convenient reference for the flux density is when θ is zero. The Lp_θ term in (12) thus simplifies to

$$Lp_\theta = 20 \log_{10} \left(\frac{r_\theta}{r_{\theta=0}} \right)_{(dB)}. \quad (17)$$

Using a nominal GPS SV orbit radius of 26,560 km and a mean Earth radius of 6,370 km, the space path loss relative to $\theta=0$ is shown in Figure 7 for users on the surface of the Earth using Equation 17.

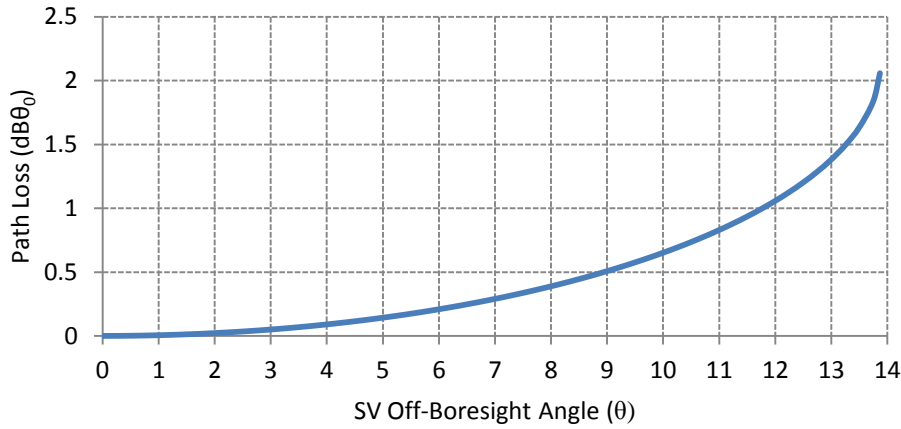


Figure 7: Path Loss of GPS signals relative to $\theta=0$

3.3.3 SV Gain Patterns

The gain pattern of the SVs are designed to compensate for the effects of path loss, such that there is more gain towards the limb of the Earth to give a more uniform power over the surface. Ideally, a pencil beam radiation pattern would provide a uniform power distribution over the Earth's surface, but this is difficult to achieve in practice. Rather, a dimpled lobe is realized with a helical antenna array that does not result in a uniform power distribution over the Earth's surface. Figure 8 depicts the ideal pattern and the approximate actual main lobe pattern of a SV antenna.

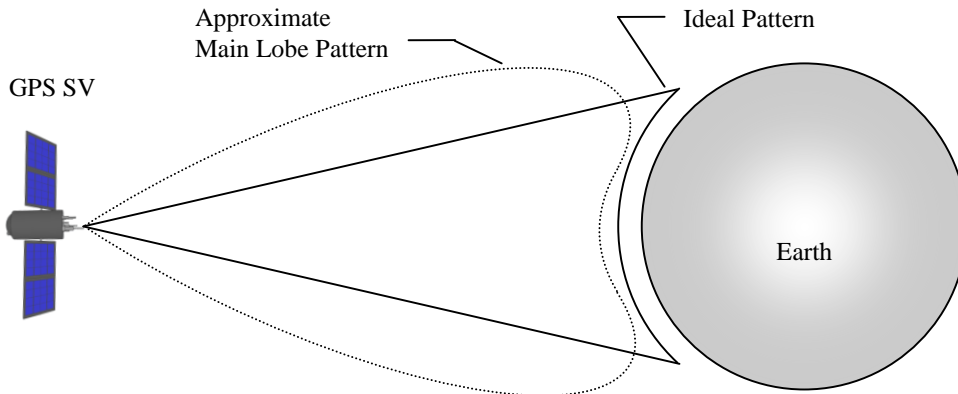


Figure 8: GPS SV Radiation Pattern

Over the years, improvements have been made to the design of the antenna array panel, resulting in differing radiation patterns for various SV generations called *blocks*. Table 2 summarizes the SV members belonging to each SV design generation. There are many differences in the features and capabilities of the SVs from one block group to the next, but the design component of interest affecting SNR levels is the radiating L-Band antenna array panel.

Table 2: SV Block Members

Block	SV Numbers	Antenna	Active*	Description
I	1-11	I	0	Original concept validation satellites
II	13-21	II	0	First full scale operation satellites
IIA	22-40	II	12	Improved autonomy satellites
IIR	41-46,51,54,56	IIR Legacy	8	Replenishment satellites for Block IIA
IIR	47,59,60,61	IIR Improved	4	
IIR-M	48,49,50,52,53,55,57,58	IIR Improved	7	Modernized replenishment satellites for Block II
IIF	Unknown	Unknown	0	Follow On Satellites to Block IIR (planned)

* As of July, 2009

The initial block I members all shared the same validation design. Blocks II and IIA have a similar design and radiation pattern to block I, which was studied in [14]. Block IIR has two types of designs, referred to as the legacy and improved panels. Ground based radiation pattern measurements were performed on Block IIR satellites in [15] and [16], while on-orbit measurements for Block IIR satellites have been performed in [17] and [18]. The measurements in [17] and [18] differentiated the antenna types only by block members, when in fact not all members of block IIR share the same design. Therefore the data from [16] is taken as more accurate for Block IIR patterns because the correct distinction is made. Figure 9 shows the radiation patterns for the three different SV antenna designs.

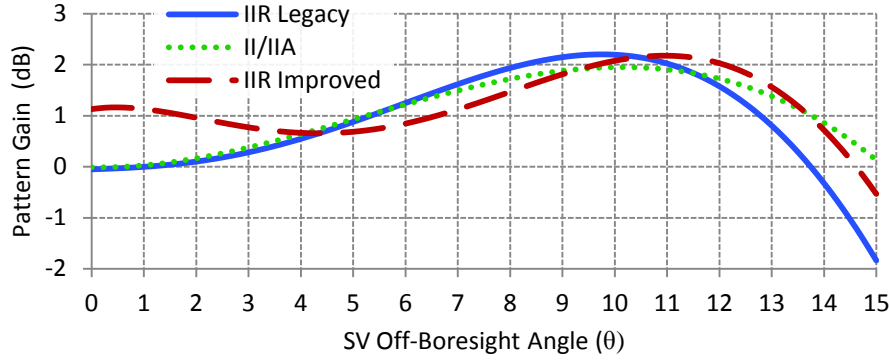


Figure 9: SV Transmit Antenna Radiation Patterns

Fifth order approximations of the patterns are provided in Table 3. To determine the effect of the transmitting GPS SV gain pattern on the received signal strength, Equation 16 is used to convert the measured SV elevation ϕ to the SV off-boresight angle θ . The polynomial coefficients in Table 3 are then used to determine the $G_{t\theta}$ term in Equation 12 for the appropriate type of SV from which the measurement originated.

Table 3: SV Antenna Radiation Pattern Coefficients (dB)

	θ^5	θ^4	θ^3	θ^2	θ^1	θ^0
II/IIA	-3.165e-19	5.748e-006	-0.0040676	0.061662	-0.019623	-0.0066788
IIR Legacy	2.5085e-5	-0.0011136	0.010073	0.0044916	0.035453	-0.051893
IIR Improved	5.8997e-5	-0.0028536	0.040936	-0.18848	0.14967	1.1324

Table 4 shows all the active SVs with the corresponding antenna type. The most up-to-date constellation status is available on the U.S. Naval Observatory website <ftp://tycho.usno.navy.mil/pub/gps/gpstd.txt>. It should be noted that the PRN assignments occasionally change, and so the U.S. Naval website should be consulted to determine the antenna type in Table 4 for the particular SV using a PRN.

Table 4: GPS Satellite Constellation

SV Block	Antenna Design	SV Number	PRN	Launch Date
IIA	II	23	32	Nov 26, 1990
IIA	II	24	24	July 4, 1991
IIA	II	25	25	Feb 23, 1992
IIA	II	26	26	July 7, 1992
IIA	II	27	27	Sept 9, 1992
IIA	II	39	09	June 26, 1993
IIA	II	34	04	Oct 26, 1993
IIA	II	36	06	Mar 10, 1994
IIA	II	33	03	Mar 28, 1996
IIA	II	40	10	July 16, 1996
IIA	II	30	30	Sept 12, 1996
IIA	II	38	08	Nov 6, 1997
IIR	IIR Legacy	43	13	July 23, 1997
IIR	IIR Legacy	46	11	Oct 7, 1999
IIR	IIR Legacy	51	20	May 11, 2000
IIR	IIR Legacy	44	28	July 16, 2000
IIR	IIR Legacy	41	14	Nov 10, 2000
IIR	IIR Legacy	54	18	Jan 30, 2001
IIR	IIR Legacy	56	16	Jan 29, 2003
IIR	IIR Legacy	45	21	Mar 31, 2003
IIR	IIR Improved	47	22	Dec 21, 2003
IIR	IIR Improved	59	19	Mar 20, 2004
IIR	IIR Improved	60	23	June 23, 2004
IIR	IIR Improved	61	02	Nov 6, 2004
IIR-M	IIR Improved	53	17	Sep 26, 2005
IIR-M	IIR Improved	52	31	Sep 25, 2006
IIR-M	IIR Improved	58	12	Nov 17, 2006
IIR-M	IIR Improved	55	15	Oct 17, 2007
IIR-M	IIR Improved	57	29	Dec 20, 2007
IIR-M	IIR Improved	48	07	Mar 15, 2008
IIR-M	IIR Improved	49	01	Mar 24, 2009
IIR-M	IIR Improved	50	05	Aug 17, 2009

(August, 2009, <ftp://tycho.usno.navy.mil/pub/gps/gpssat.txt>)

3.3.4 Atmosphere

Although not very significant, there is some attenuation of the GPS signals by the atmosphere. Atmospheric attenuation is not a concern for a GPS receiver in space, but for the purpose of making ground measurements the effect is examined. Signals arriving from the zenith will encounter almost no attenuation, while signals arriving near the horizon will travel through a greater distance of the atmosphere and be attenuated more. The effect on the ground is approximated by the equation [14]

$$\text{Loss(dB)} = \frac{0.07(1+2a)}{\left(\sin \phi + \sqrt{\sin^2 \phi + 2a + a^2}\right)}, \quad (18)$$

where $a = H_m/R_E$; H_m is the equivalent height for oxygen, $H_m = 6$ km, and R_E is the Earth radius, where $R_E \approx 6370$ km. Figure 10 shows the attenuation from Equation 18, with only very low elevations having any significant attenuation. The atmospheric attenuation correction La in Equation 12 is applied to the measured SNR value according to Equation 18.

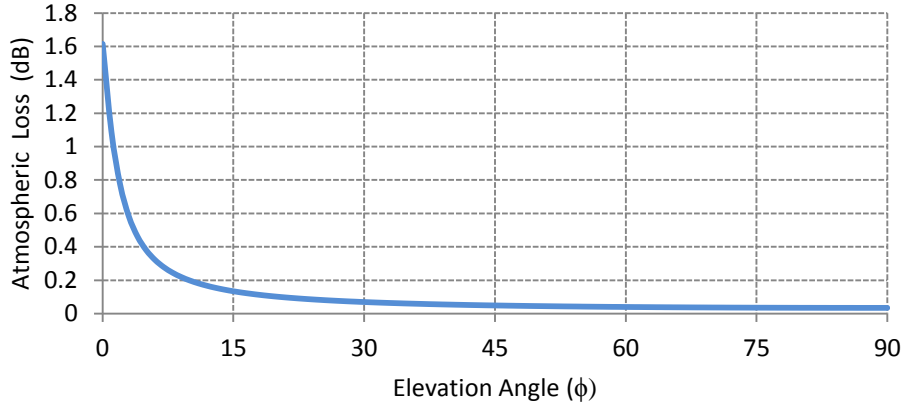


Figure 10: Atmospheric Attenuation at the Earth's Surface

3.3.5 SV Yaw Attitude

The GPS SV has two attitude constraints: the antenna boresight must remain pointing at the center of the Earth, and the SV solar panels need to remain perpendicular to the Sun. The SV yaw attitude therefore must change to maintain these constraints. In other words, the SV remains pointing directly at the Earth while rotating to optimize the solar panel orientation. This is illustrated in Figure 11.



Figure 11: SV Yaw Attitude Changes to Optimize Solar Panel Orientation in Orbit

This yaw rotation would not affect the SNR measurements if the SV radiation pattern were perfectly symmetrical, but this is not the case. Asymmetries in the azimuth direction slightly changes the power distribution on the Earth as the yaw is changed. The yaw attitude changes relatively slowly during most of its orbit, except for the so called noon turn and midnight turn maneuvers where a much higher yaw rate occurs to maintain the optimal yaw attitude [19].

The SV yaw attitude can be modeled for each SV, but this involves significant additional complexity and more information is needed about the SV transmitting antenna array than is available in the public knowledge domain. Therefore, a correction is not applied to the measured SNR value relating to the SV yaw attitude. Instead, the mean radiation pattern in the azimuth direction is used. Asymmetries in the radiation pattern contribute to error of the SNR measurement.

3.3.6 Ionospheric Scintillation

As with the atmospheric attenuation, ionospheric scintillation is not a concern for a GPS receiver in space, but because it potentially affects the ground tests, it deserves examination. Ionospheric scintillation is defined as amplitude and phase signal fluctuations caused by irregularities of the electron density in the ionosphere [13]. The effects of scintillation are sufficient enough to cause a GPS receiver to completely lose a GPS signal for short periods of time, but generally all of the visible SV signals are not affected simultaneously. The ionospheric irregularities are most

pronounced during the peak of the 11-year solar cycle, with the equatorial regions normally experiencing the phenomenon shortly after sunset, and higher latitude regions generally experiencing weaker but more prolonged periods of scintillation [13]. Modeling the effects of scintillation is difficult, and not practical for real time embedded applications. Some information about the occurrence of scintillation is however possible if a dual frequency GPS is used. The degree to which a signal is affected by scintillation is dependent upon frequency, and thus the effects can be extracted by comparing the signal phase and amplitude at the different L1 and L2 frequencies. Dual frequency low cost COTS are however not available at the time of this writing. The effects of scintillation are not factored into the calculations in this study, thus contributing to the overall error of the SNR measurements.

3.3.7 Multipath

Signal multipath occurs when an antenna receives signals from the same source via multiple paths, usually because of reflections. Reflections are caused by the Earth or any other objects near the receiver and is a significant error contributor to GPS measurements on the ground, both for position and attitude determination. Reflections cause the SNR value to fluctuate and is more likely to occur for signals arriving at lower elevations for ground-based measurements. An orbiting CubeSat has low multipath susceptibility, unless it has protruding objects such as deployable solar panels that could reflect GPS signals to the antenna under various conditions.

The effect of multipath for GPS signals is reduced by using circular polarization of the signals. The GPS signals use right hand circular polarization (RHCP). A reflected signal changes polarization direction, meaning signals that are reflected any odd number of times are left hand circular polarized (LHCP). Most GPS antennas have some degree of LHCP signal rejection. The effects of multipath are not calculated in this study, and so contribute to the noise and error of the SNR measurements.

3.4 Antenna Characterization

The angle at which a GPS signal arrives at the receiving antenna affects the signal strength (and hence SNR) of the signal at the input of the GPS receiver. The radiation pattern of the antenna influences the relationship of SNR versus the angle α . A hardware calibration mapping is performed to characterize the SNR-to- α relationship.

There are three basic ways to characterize an antenna so that it is calibrated for the SNR-to- α mapping. If the antenna is a commercial item, the radiation pattern may be extracted from a datasheet. An anechoic chamber may also be used, where measurements can be taken at whatever detail is desired. Otherwise, GPS signal measurements can be taken using a precisely known static attitude and LOS SV vectors. A good quality antenna characterization is possible if the measurements are averaged over many hours and include data points from wide range of elevations.

3.5 GPS Receivers

The choice of which COTS GPS receiver to use is separated into two groups: Ground based testing and space application. The criteria for each purpose are mostly the same, with the space application receiver having the additional and all important requirement to be able to function in orbit. The remaining criteria will be considered equivalent for both ground and space applications. Some of the criteria are impossible to fulfill simultaneously, and so design trade-offs must be made.

3.5.1 Space Capability

Under the COCOM agreement between various nations (now succeeded by the Wassenaar Arrangement), GPS manufacturers place operational limits on COTS products for security reasons. The GPS receiver will stop working when both 515 m/s velocity and 18,300 m altitude are exceeded [8]. This presents a major obstacle for budget conscious CubeSat missions attempting to use a COTS GPS product.

Another consideration for space applications is the physical environment. Certain electronic devices such as aluminum electrolytic capacitors are prone to failure

in vacuum conditions. Being aware of such environmental limitations is important when using COTS components.

3.5.2 Low Power

The power budget for small satellites is very restrictive. The small size of the satellite solar panels limits average power consumption to a few watts at best for the entire system. Intelligent duty cycling of the satellite subsystems is normally required to conserve energy. Since there is always some overhead associated with a GPS receiver establishing a signal lock when it is powered up before it becomes operational, the GPS system is somewhat limited by how much duty cycling is possible. For ground testing, low power is also a requirement since the testbed is deployed for many hours at a time for performance evaluation and data logging.

3.5.3 SNR Measurement Resolution

The National Marine Electronics Association (NMEA) defines a communication protocol that is widely used amongst GPS receiver manufacturers. A typical NMEA GPS data packet contains reports of SV SNR values in the units dB-Hz. Some manufacturers also use their own protocol such as Trimble Standard Interface Protocol (TSIP) with the SNR reported in alternate units. The SNR value is normally given as an integer data type, because very accurate measurements are not needed for typical GPS applications. A very weak GPS signal is in the range of 30 dB-Hz, while a very strong signal is in the range of 55 dB-Hz. This type of coarse measurement is not ideal for SNR attitude determination, as there are only about 25 unique quantization levels to distinguish the full range of off-boresight angles.

GPS products manufactured by Trimble are all capable of using the TSIP Protocol. This protocol allows for the SNR units to be provided in either dB-Hz units as integers, or AMUs as floating point numbers. The resolution of the AMU units is not consistent for all Trimble products, but the resolution has been found to be better than the dB-Hz units for the Trimble Copernicus GPS receiver listed in Table 5. The relationship of the SNR between dB-Hz and AMUs is [20]

$$dB \cdot Hz = 10 \log_{10}(BW \times AMU^2), \quad (19)$$

where BW is the measurement bandwidth, which is 1 kHz in Trimble products.

3.5.4 Update Rate

The standard update rate for most COTS GPS receivers is 1 Hz. Recently a number of GPS receivers have become available that can now provide update rates of 5 Hz or even 10 Hz. The attitude of a small satellite is not normally changing so quickly that such that a high update rate is required, but the additional samples could be beneficial for improving the quantization level via oversampling.

3.5.5 GPS Receiver Choices

Four GPS receivers are considered for use with the testbed and future implementation on a CubeSat. Table 5 summarizes the features of each module relating to the most important selection criteria.

Table 5: GPS Receiver Features

GPS	Space Capable	SNR Units	Update Rate	Power Consumption	Mass
Trimble Copernicus	No	dB-Hz / AMU	1 Hz	120 mW	1.7 g
SkyTraq Venus634FLPx	No	dB-Hz	10 Hz	100 mW	1.0 g
DLR Phoenix	Yes	dB-Hz	1 Hz	850 mW	20 g
Navman Jupiter 12	Tested Beyond COCOM Limits	dB-Hz	1 Hz	280 mW	25 g

The Trimble Copernicus GPS receiver is chosen for ground based testing of the SNR ADS because it is able to provide a higher resolution SNR measurement in AMU units rather than dB-Hz units. This allows for more equal performance comparisons to the previous results which also used Trimble products. Although the Copernicus receiver is not a space capable module, it serves as a good module for ground measurement testing.

The SkyTraq Venus634FLPx is considered because of its low power consumption, high update rate of 10 Hz, and extremely small size of 1 cm². To

perform ground measurements, the SkyTraq is probably the best candidate for modules with dB-Hz SNR measurement units.

The German Space Agency (DLR) has developed the Phoenix space borne GPS based on the Orion architecture. This model of GPS has been implemented in the Compass-1 CubeSat design [2] which was successfully launched on April 28, 2008. The flight tested design in a CubeSat application make this GPS receiver a very good candidate. Obtaining such a specialized receiver may be difficult.

The Navman Jupiter 12 GPS was used on the payload of the sounding rocket SRP-5 mission at the University of Alaska in January, 2009. The GPS provided raw data packets during the flight, which encountered conditions far exceeding the COCOM limits. Further testing of this GPS module may reveal satisfactory performance for orbital environments.

Chapter 4: Antenna Design

4.1 Conventional GPS Antennas

One of the most widely used antenna for GPS applications are microstrip patch antennas. Helical antennas are also sometimes used for applications such as smart phones. Other specialty antennas such as choke-ring patch antennas are used for precision geodetic surveying applications.

Because the measurements of SNR attitude determination depend directly on the gain pattern of the receiving antenna, it is the characteristic that is most important for SNR attitude determination. The reasons are as follows: Consider an isotropic antenna. The gain is equal in all directions. There would be no variation in signal strength regardless of the antenna attitude. This type of antenna would not reveal any information about the angle of the arriving signal. On the other extreme we have a parabolic antenna. The gain is very high, but must be precisely aimed at the transmitting signal. Adjusting the antenna attitude slightly would result in a large change in receive signal SNR. This is good for accuracy, but the receive signal will be too weak for detection in all cases except for the unlikely event that the antenna is pointed directly at the transmitting SV. In GPS applications, signals are used from multiple SVs simultaneously, originating from multiple directions. Thus, a compromise is needed between accuracy and field of view. The goal is to have a significant signal strength variation from the boresight to 90° , but still have sufficient signal strength at 90° for the GPS receiver to detect the signal. An initial estimated target is for the pattern amplitude roll-off to be about 20 dB from the boresight to 90° . This estimation comes from the observation that GPS receivers are able to decode signals reliably at about 35 dB-Hz, while the highest observed signals are in the range of about 55 dB-Hz. If the gain pattern amplitude roll-off is steeper than 20 dB / 90° , it is more likely that GPS signals will be out of the antenna's field of view, and accuracy will be diminished with fewer GPS SV signal locks. On the other hand, if the roll-off is too shallow, there will not be a significant difference in SNR when the off-boresight angle is changed, thus reducing the resolution of the antenna mapping function.

The pattern amplitude roll-off should also ideally be monotonic with minimized ripple. If there is ripple in the radiation pattern, or if there are side lobes, then there will be non-unique off-boresight angles for any given received signal strength. The radiation pattern should also be symmetrical about the azimuth direction. This is because the attitude algorithm is based on the dot product of the antenna boresight vector and SV LOS vector. It makes no distinction for an azimuth variation.

The following four antenna types will be examined to determine the best candidate for SNR attitude determination: conventional microstrip patch, choke-ring patch, pinwheel, and shorted annular patch.

4.1.1 Conventional Microstrip Patch Antenna

The conventional microstrip antenna is often used for automotive and aviation applications. Both rectangular and circular radiating elements are common. The theoretical gain pattern of the conventional patch resembles a hemispherical lobe, assuming an infinite ground plane. This is ideal for normal GPS applications, because the entire sky is within a zenith pointing antenna's field of view. The antenna is able to receive signals from all areas of the sky at about the same signal strength.

The theoretical gain pattern is not realized however because real antennas do not have infinite ground planes. The radiation pattern is significantly affected because of E-field diffraction at the edge of the ground plane [21]. Figure 12a illustrates the difference in the E-field radiation patterns between the theoretical and measured pattern of a conventional patch antenna. The ripple in the pattern is of no consequence for the typical GPS user as long as the SNR is high enough to detect the signal. For GPS SNR attitude determination measurements however, the radiation pattern ripple is problematic. The mapping of off-boresight angles to signal strength becomes non-unique. Also, the gain roll-off is shallow, resulting in little appreciable signal strength difference from the boresight to 90° .

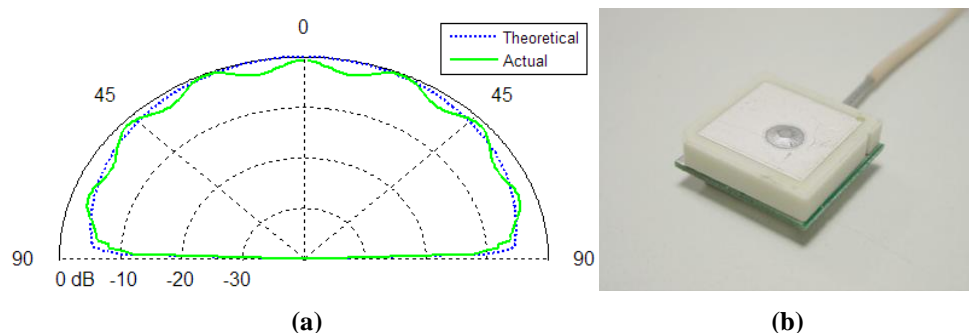


Figure 12: Conventional Patch Antenna (a) Example of Theoretical and Actual Radiation Pattern Comparison, and (b) Miniaturized High Permittivity Dielectric Square Patch

4.1.2 Choke-Ring Patch Antenna

The choke-ring antenna consists of a normal patch antenna residing on a concentric corrugated ground structure and is categorized as a Reduced Surface Wave (RSW) antenna. The primary function of the choke-ring, shown in Figure 13 is to minimize accuracy-reducing multipath signals. A side effect of the choke ring is a gain pattern amplitude roll-off that is steeper and smoother than the conventional patch, which are a desirable characteristic.

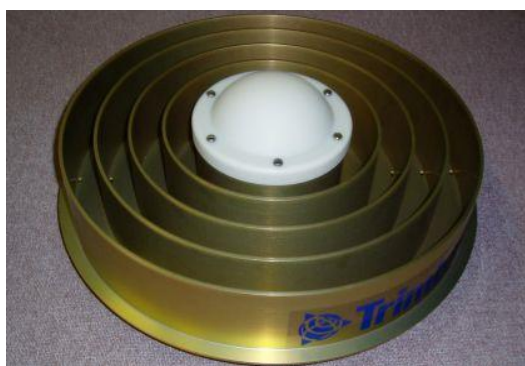


Figure 13: Choke-Ring Antenna.

Source: www.ngs.noaa.gov/ANTCAL/

Multipath signals are rejected because the antenna gain near and below the horizon is suppressed by the choke-ring. The majority of multipath signals arrive from these regions, either from ground reflections or by reflections off objects on the

ground. High precision applications such as surveying or carrier wave interferometry benefit from using a choke-ring antenna. Despite the desired electrical characteristics, size and mass exclude a choke-ring antenna for small satellites.

4.1.3 Pinwheel Antenna

The pinwheel antenna is a microstrip RSW design. A carefully crafted spiral pattern with outer concentric rings is etched as the radiator. Substrate air gaps line the perimeter of the circular antenna as shown in Figure 14. The characteristics of the pinwheel were measured in [22] and found to be similar to the choke-ring. The gain amplitude roll-off is about 15 dB from the boresight to 90°. The light weight and radiation pattern characteristics of the design are favorable, however the complexity involved in manufacturing the antenna prevented its use in this study.

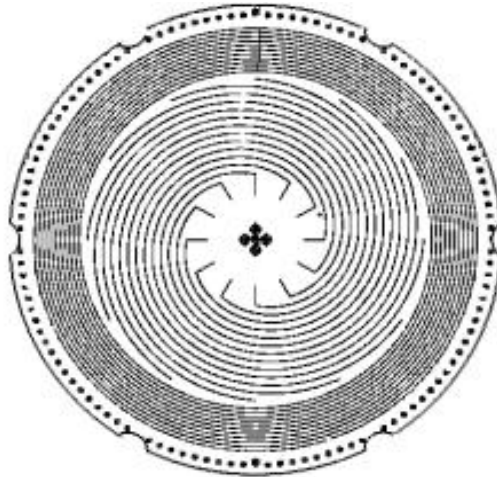


Figure 14: Spiral Pinwheel Antenna Structure
(Source: Kunysz, NovAtel Inc.)

4.1.4 Shorted Annular Patch Antenna

The SAP antenna is also a RSW antenna like the choke-ring, but its mass and size is similar to a conventional patch antenna. Because the surface waves of the antenna are reduced, the ground plane edge diffraction is also reduced. The result is a better agreement between the theoretical and measured radiation pattern [23]. An example is illustrated in Figure 15. The gain pattern roll-off is also significantly steeper than the conventional patch—a desired characteristic for both multipath reduction and SNR mapping function resolution. The SAP antenna was shown in [22] to provide similar or better performance than the choke-ring design while being at

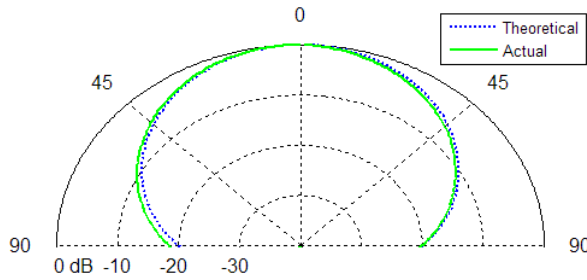


Figure 15: Theoretical and Actual Radiation Pattern for a SAP Antenna

least an order of magnitude smaller in terms of volume, and being smaller and easier to manufacture than the pinwheel design. The improved performance and low mass make the SAP antenna a strong candidate for use with SNR attitude determination.

4.2 Shorted Annular Patch Antenna Theory

The principle of the SAP antenna is that surface waves and lateral radiation is minimized if the outer radius is set to meet the condition [24]

$$a = \frac{x'_{11}}{\beta}, \quad (20)$$

where x'_{11} is the first root of the derivative of the first-order Bessel function, and β is the free-space propagation constant and a is the outer radius. The reduction of lateral radiation consequently reduces ground plane edge diffraction which contributes to the non-monotonic radiation pattern of conventional patch antennas. The condition in Equation 20 results in a larger radius compared to a conventional patch antenna, and the antenna is no longer resonant for the desired frequency without further adjustment. To make the antenna resonant, a short circuited annular ring is introduced at the center

of patch with radius b . The frequency of the SAP antenna operating in the TM_{11} mode relating to the outer and inner patch radii is given by [24]

$$J_1'(ka_e)Y_1(kb) - J_1(kb)Y_1'(ka_e) = 0 \quad , \quad (21)$$

where the prime notation denotes the first derivative, the subscript denotes the order of the function, J is the Bessel function of the first kind, Y is the Bessel function of the second kind, a_e is the effective outer radius, b is the inner radius, and k is the substrate wave number. Figure 16 illustrates the antenna geometry.

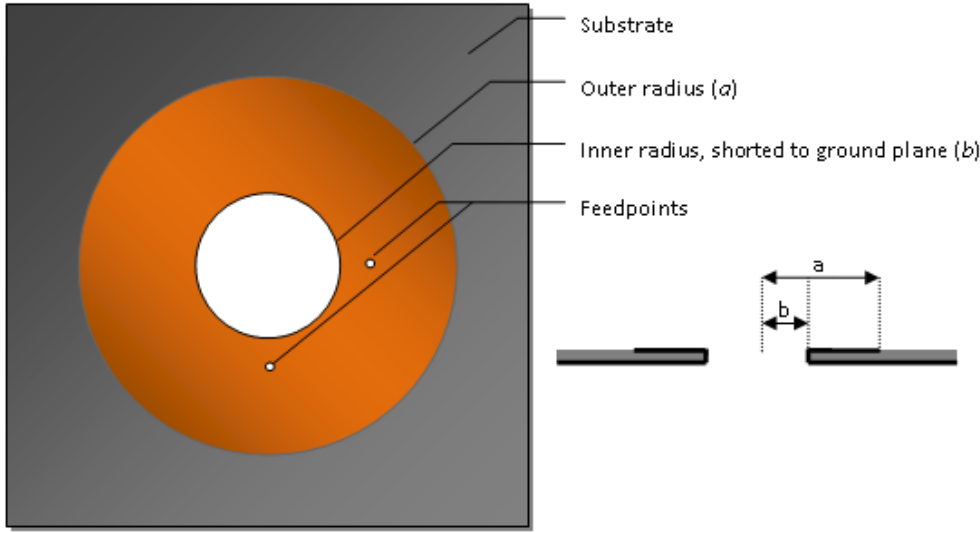


Figure 16: SAP Antenna Geometry: Top and Lateral Views

A correction is applied to the outer physical radius a to account for field fringing [24], resulting in an effective radius a_e given by

$$a_e = a \sqrt{1 + \left(\frac{2h}{\epsilon_r \pi a} \ln \left(\frac{\pi a}{2h} \right) + 1.7726 \right)}, \quad (22)$$

where h is the substrate thickness.

Unlike the conventional patch, the gain pattern amplitude roll-off is adjustable for a SAP antenna. The maximum gain and maximum surface wave reduction is achieved by the outer radius condition in Equation 20, but the outer radius may be decreased so long as the inner radius is also adjusted to satisfy Equation 21. As the

inner radius in Equation 21 approaches zero, the antenna characteristics will approach that of a conventional circular patch antenna. For the GPS L1 frequency of 1.57542 GHz, the outer radius a is 55.7 mm for maximum gain.

SAP antennas with outer radii 35, 45 and 55.7 mm were configured in [22] to determine the multipath signal rejection performance relative to other antennas. The gain pattern amplitude roll-off values were 15, 20 and 25 dB respectively. The 45 mm outer radius antenna was the best performer, even against a commercial high precision surveying antenna. The effect of the polarization axial ratio on multipath rejection was also studied in [25] for various configurations of SAP designs. The result showed that the highest surface wave reduction according to Equation 20 is not the best performer for multipath reduction. Instead, a SAP antenna with a slightly smaller outer radius than 55.7 mm has a better axial ratio and overall multipath performance. This finding is consistent with the results presented in [26].

4.3 SAP Antenna Design

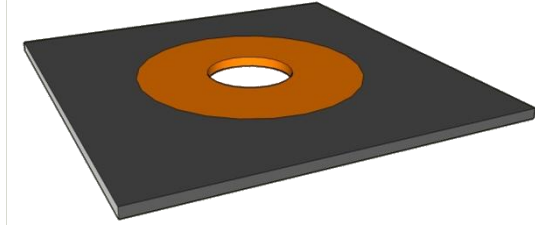
The electrical design specifications for a GPS L1 antenna are as follows:

- L1 frequency of 1575.42 MHz.
- 50 Ω Impedance.
- Right Hand Circular Polarization (RHCP).
- Bandwidth at least equal to the GPS L1 spread spectrum: 2 MHz for C/A code, measured to sinc first nulls.

The SAP antenna depicted in Figure 17 is fabricated on a milling machine using Rogers RT/Duroid 5870 substrate. Table 6 shows the design parameters. Two probe feed points are placed at a distance ρ separated by 90° to achieve right hand circular polarization and a 50 Ω impedance match. The feedpoints on the antenna are two U.FL type surface mount connectors mounted on the underside groundplane. A feed probe is soldered to the center conductor of the connector and protrudes through the dielectric to the top radiator where it is also soldered. A Mini-Circuits QCN-19 surface mount quadrature hybrid is used to sum the feed points.

Table 6: SAP Design Parameters

Parameter	SAP Design
Groundplane	150 ² mm
Inner Radius b	17 mm
Feedpoint ρ	14 mm
Outer Radius a	45 mm
Substrate Dielectric ϵ_r	2.33
Substrate Thickness h	3.2 mm

**Figure 17:** SAP Antenna

4.4 SAP Antenna Measurements

A Hewlett Packard E8803A PNA Vector Network Analyzer was used to perform the antenna electrical measurements. The calibration of the instrument was performed by a custom calibration board. The board had open, shorted and 50 Ω loads terminated directly on the same type of U.FL connector as the antenna, using the same type of substrate. This created a calibration reference to the antenna ground plane.

4.4.1 SAP Electrical Measurements

The electrical parameters were virtually identical for each feed probe, so the results are presented as representing each of the feed points. The first measured parameter is the VSWR shown in Figure 18. The antenna match looks very good at the GPS L1 frequency of 1.57542 GHz, with a VSWR of 1.067. The antenna bandwidth, defined as $SWR < 2$, is from 1.56 GHz to 1.59 GHz, or 30 MHz. The bandwidth satisfies the minimum 2 MHz requirement. The same marker locations are used from Figure 18 through Figure 20.

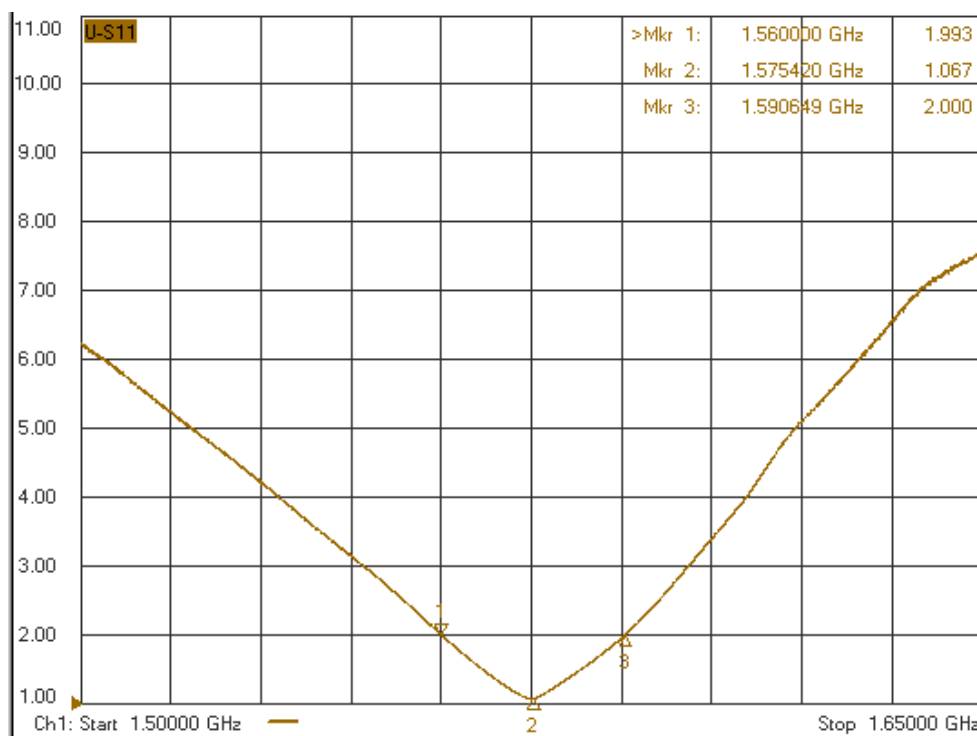


Figure 18: SAP Network Analyzer VSWR S11 Measurements

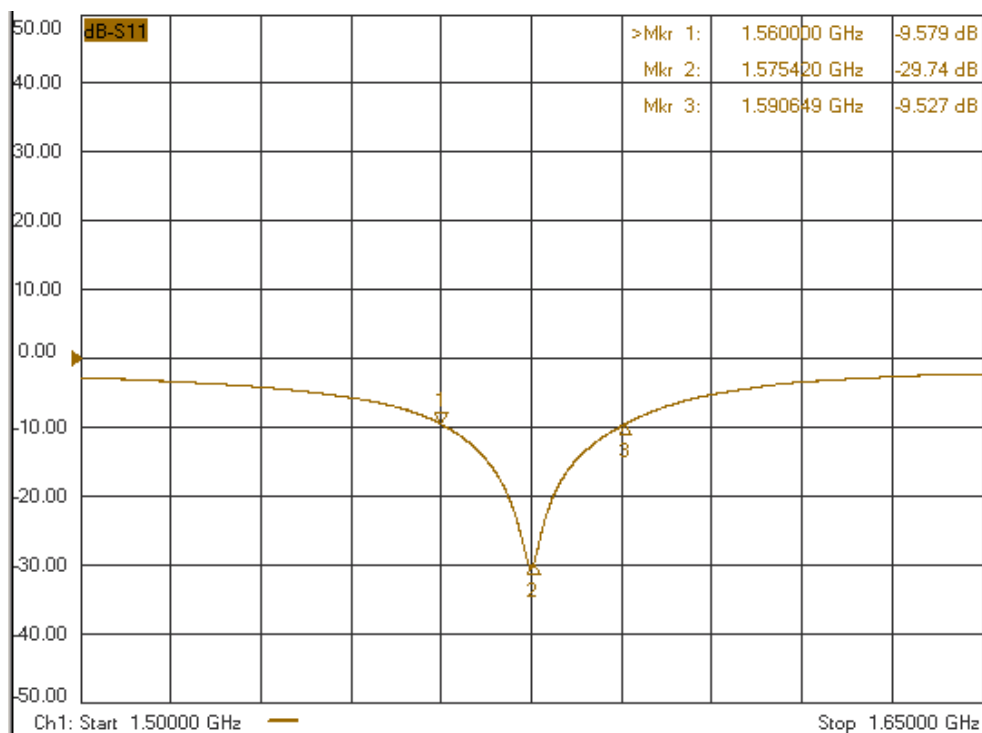


Figure 19: SAP Network Analyzer S11 Insertion Loss Measurements

The smith chart in Figure 20 confirms a well tuned antenna. The impedance is slightly capacitive at the target L1 frequency, and is closely matched to the target 50 Ω impedance at 51.67 Ω .

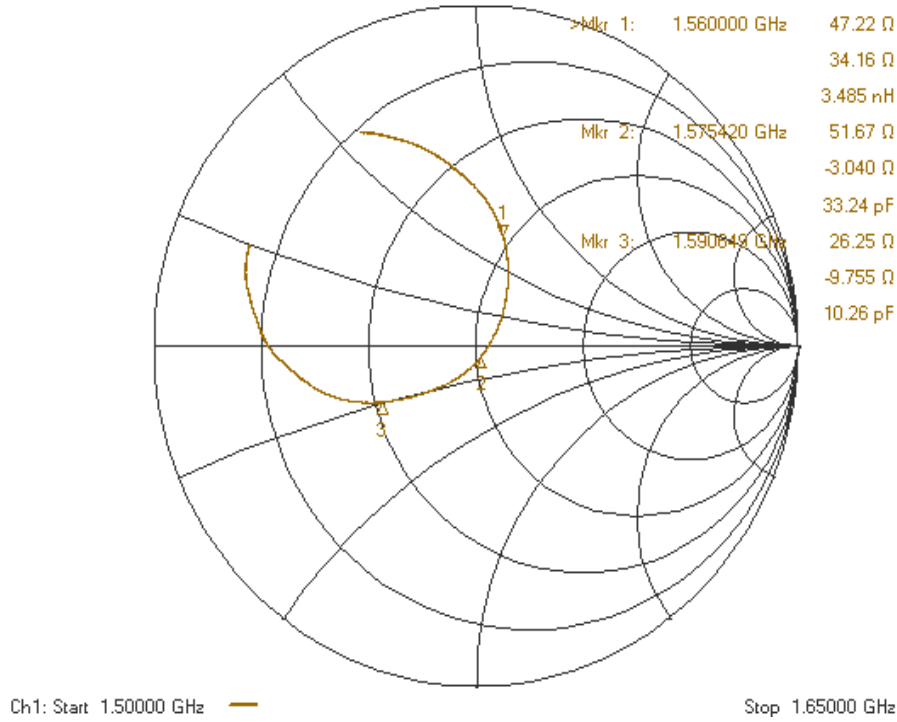


Figure 20: SAP S11 Smith Chart Measurement

4.4.2 SAP Radiation Pattern Measurements

The radiation pattern was not measured directly in an anechoic chamber, but was rather inferred by measuring the SNR of GPS SVs over a period of time, similar to the procedures used in [3], [4], [5], [6], [7]. The SAP antenna was statically set up in an open field in the zenith direction for approximately 20 hours. Over this time, a good range of elevation and azimuth data points were collected in order to construct the antenna radiation pattern.

The SNR measurements in Figure 21 were collected as linear AMU units using a Trimble Copernicus GPS. A best-fit curve is modeled from the SNR which represents the SNR-to- α mapping function.

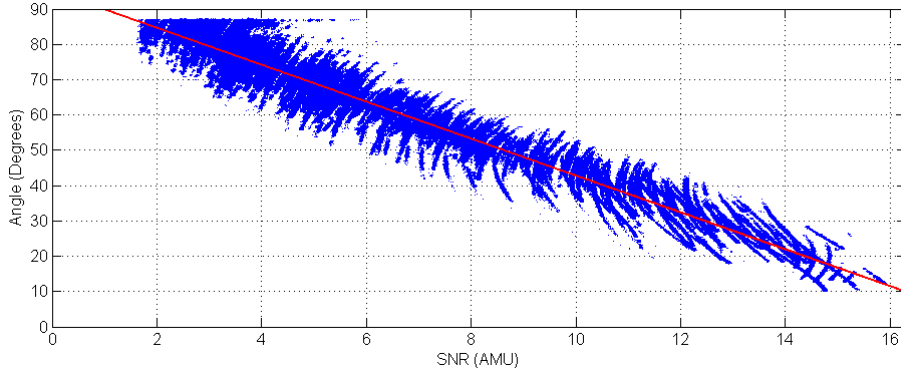


Figure 21: SNR-to- α Mapping: Measured AMU Samples (Blue) and Best Fit (Red)

The coefficients of a third order polynomial mapping function are listed in Table 7 where the SNR values are in AMU units, normalized to the maximum AMU level so that

$$SNR_N = \frac{SNR}{k}, \quad (23)$$

where $0 < SNR_N < 1$ and the maximum measured SNR is k .

Table 7: SAP Antenna Mapping Function Coefficients

SNR_N^3	SNR_N^2	SNR_N^1	SNR_N^0
-42.3644	68.6685	-118.4877	99.7649

The function is converted to dB-Hz units using Equation 19 and plotted in Figure 22, and is then used to generate the standard polar radiation pattern in Figure 23. The radiation pattern and SNR-to- α mapping are inverse functions of each other. There is no distinction for any variations in the azimuth direction that may exist in the pattern.

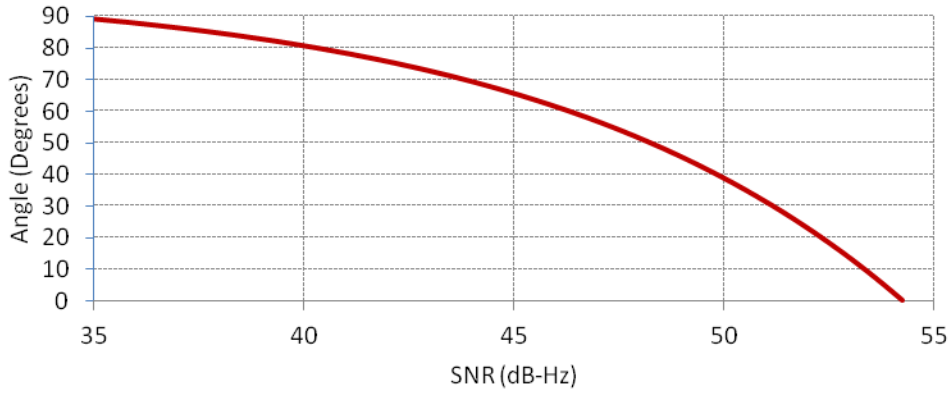


Figure 22: SNR-to- α Mapping Best Fit from Figure 21 in dB-Hz Units

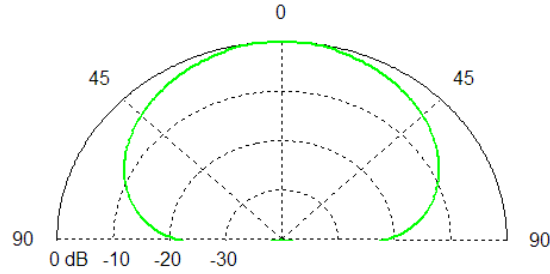


Figure 23: SAP Antenna Radiation Pattern from Best-Fit in Figure 22

The axial ratio was not measured for the antenna, but should be measured in any future work. The measurements in [25] are a good comparative reference for a SAP antenna with the same dimensions as described in Table 6. Anechoic chamber measurements should also be performed to confirm the radiation pattern in Figure 23 and test for any azimuth variations.

Chapter 5: Testbed Design

5.1 Design Overview

The testbed is an embedded system that serves as a development and validation platform that can be used as a reference design for a SNR satellite attitude determination subsystem. A simplified block diagram of the testbed hardware design is shown in Figure 24.

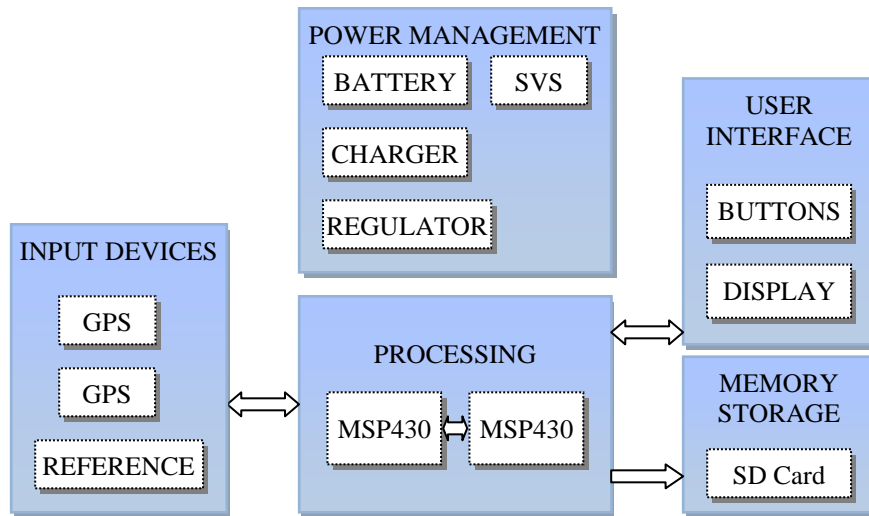


Figure 24: Testbed Block Diagram

The processing hardware is comprised of two Texas Instruments MSP430 microcontroller Units (MCUs). Two MCUs are used in the design for maximum flexibility and capability of the testbed system. The alternative approach is to use a more capable processor such as an ARM Cortex-M series MCU, however such processors have fewer low power modes and flexibility compared to the MSP430 series. For example, in order to use certain peripherals such as UARTs, the Cortex-M series processors must remain in a high power mode even while the processor is idle. By contrast, the MSP430 series MCUs have more fine-grained power control that allows peripherals to be operational while the MCU operates in deep power down modes. Using two processors allows the power consumption of the overall system to scale proportionally to the processing requirements. The power consumption stays

low when the processing demands are low. As the processing demands increases beyond the capability of a single processor, the additional processor is capable of meeting the added processing requirement. The dual processor approach also increases the available memory and total number of peripherals, allowing more input devices to be attached to the system. A newly developed protocol is used for inter-processor communication. More detail about the protocol is discussed in more detail later in the chapter.

The input devices include at least one GPS Receiver, but a second GPS Receiver may also be used. The optional second GPS Receiver allows the testbed to:

- Simultaneously compare different antenna types.
- Simultaneously compare different GPS Receivers.
- Use two non-aligned antennas to calculate a 3-axis attitude estimation.

A reference may also be connected as an input device. It serves as a truth attitude estimation comparison and can be any type of device such as a magnetometer or carrier wave attitude GPS system.

A removable micro SD Card serves as the storage medium that logs GPS data for analysis and post processing the data at a later time. This allows for the testbed to be a stand-alone embedded system— not requiring any computer connection while measurements are being made.

The user interface consists of push buttons and an Organic Light Emitting Diode (OLED) graphical screen. The screen displays the control menu and other selectable parameters. The user interface allows the user to easily configure and operate the testbed according to the desire of the user.

Power is supplied either by a 3.7 V Polymer Lithium Ion battery or an external 5V supply via a barrel connector. The external supply provides power when connected, but if the external supply is not available the power is supplied by the battery. A step-up switching regulator is used to boost the input voltage to 15V for the OLED screen, and a 3.3V linear regulator with low quiescent current is used for all the IC supplies. A Supply Voltage Supervisor (SVS) peripheral on the MSP430 MCU

serves as an under voltage lockout comparator to avoid over discharging the lithium battery.

Several revisions of the testbed were designed and manufactured during the course of hardware development: Rev. 1.2, Rev. 2.0 and Rev. 2.1. Table 8 summarizes the main design differences between board revisions. Figure 25 shows the circuit boards of the different design revisions.

Table 8: Testbed Hardware Revision Differences

Hardware Revision	Clock	Inter-Processor Communication	GPS	Flash Storage	MSP430 MCUs
1.2	DCO	8 bit I/O	On board	On board Flash	2 x F249
2.0	Crystal	8 bit DMA	External	Removable SD Card	1 x F5419 1 x F2617
2.1	Crystal	16 bit DMA	External	Removable SD Card	2 x F2617

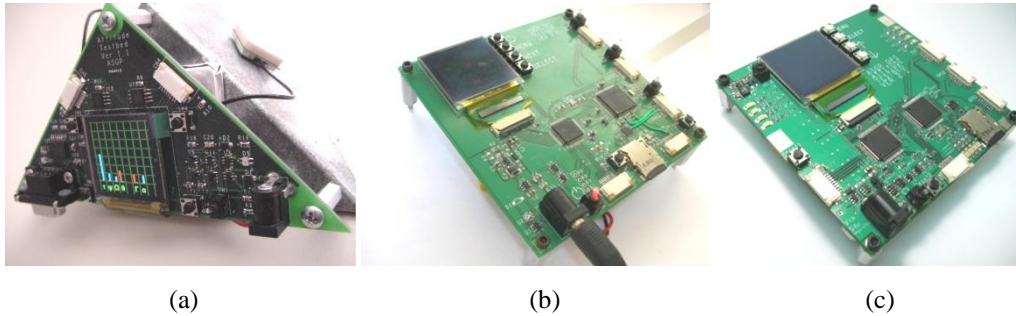


Figure 25: Testbed PCBs of (a) Revision 1.2, (b) Revision 2.0 and (c) Revision 2.1

5.2 Hardware Revision 1.2 Design

Revision 1.2 was the first PCB that was manufactured. The board performed well for the initial design, and software was developed to successfully compute a real-time attitude solution. Nevertheless, the board did have some shortcomings that needed attention. The board did not have an external crystal and used only the Digital Controlled Oscillator (DCO) for a clock source. The internal DCO has a much lower temperature stability compared to a crystal, and this became problematic during very low temperature testing when the system became unreliable because of serial communication port clock skews.

The Rev 1.2 PCB design had another constraint which was to have the GPS receiver module connector header directly on the PCB. This limited the design to be capable of using only one kind of GPS receiver module. A more versatile design adopted in later hardware revisions was to have the GPS module on a small separate PCB with a ribbon cable connection to the testbed. This allows any GPS module to be used with the testbed.

Lastly, the storage solution for Rev 1.2 was to have two SPI flash memory ICs. Data stored to flash would need to be download via a serial cable to a host computer to be analyzed. Although this was a workable system, removable SD Card flash media was used in later revisions for more convenience and much higher storage capacities.

5.3 Hardware Revision 2.0 Design

Some of the shortcomings of Rev 1.2 were addressed with the design of Rev 2.0 hardware. A crystal oscillator was used to solve the communication problems at extreme temperatures. A new Inter-Processor Communication (IPC) protocol was developed for faster communication using Direct Memory Access (DMA) controllers and timer peripherals. The GPS receiver modules were no longer connected directly to the testbed to allow more flexibility of using various types of receivers. The newest available MSP430 variant 54xx series MPU was also used for the higher available clock speed, more communication ports and memory capabilities. However, the decision to use the newest hardware available proved to be troublesome due to some compatibility issues with the software development tools.

5.3.1 MSP430 54xx Series Issues

Compatibility issues became apparent when development began with the newly added 54xx series MCU. Considerable time was spent troubleshooting erroneous program results before the following problems were discovered: The watchdog timer and hardware multiplier memory address definitions were incorrect in the software tools, and a hardware bug caused incorrect program flow while using the Joint Test Action Group (JTAG) debugger.

Normally when a MPU peripheral address such as a watchdog is not the same from one MPU variant to the next, it is transparent to the programmer because symbolic references are used in a header file of the program. The problem in the case of the 54xx arose because all previous variants of the MSP430 used a common address for both the watchdog timer and the hardware multiplier. The programmers of the Pumpkin RTOS took this for granted and hard-coded the addresses in the RTOS library. Thus, when the code from the library was executed on the 54xx MPU, sporadic resets would occur when the watchdog timer peripheral was accessed from within the RTOS library code. After consulting with the RTOS vendor, the problem was confirmed and the vendor agreed to provide a new library that did not use the watchdog timer. While this solved the sporadic reset problems, unexpected program results persisted during code development. It was later found that the Imagecraft compiler also used an incorrect peripheral address for the hardware multiplier. The hardware multiplier peripheral was disabled within the software tools as a work-around, but this also reduced processing performance.

While running the 54xx hardware in the debug mode in an attempt to discover a software bug, the program executed in an unexpected fashion. When the MCU errata sheet was closely examined, it was discovered that a hardware bug causes the MCU execute code at the incorrect address while using the JTAG debugger (errata item EEM6). The errata workaround was to manually set the program execution address if continuing to executing code after reaching a debugging breakpoint.

Fearing even more time-consuming obstacles would arise that could prevent timely progression of software development, the 54xx series MCU was abandoned and a subsequent PCB was designed to use the 26xx series MCU which was known to be compatible with all the software tools. The compiler tools were also changed from Imagecraft ICC to Texas Instruments Code Composer Studio. The lesson learned here is to be familiar with all silicon errata pertaining to the design early in the design phase.

5.4 Hardware Revision 2.1 Design

The overall design of Rev 2.1 was kept nearly the same as Rev 2.0 with the exception of eliminating the 54xx series MPU in favor of the proven 26xx series. The IPC protocol was also changed to a 16-bit data bus from 8-bit. Some other minor enhancements included adding more status LEDs for debugging, a header for power measurements, ground pads for oscilloscope probes, and configuring the power management for soft shutdown control. A detailed block diagram is shown in Figure 26 for the final design of the testbed.

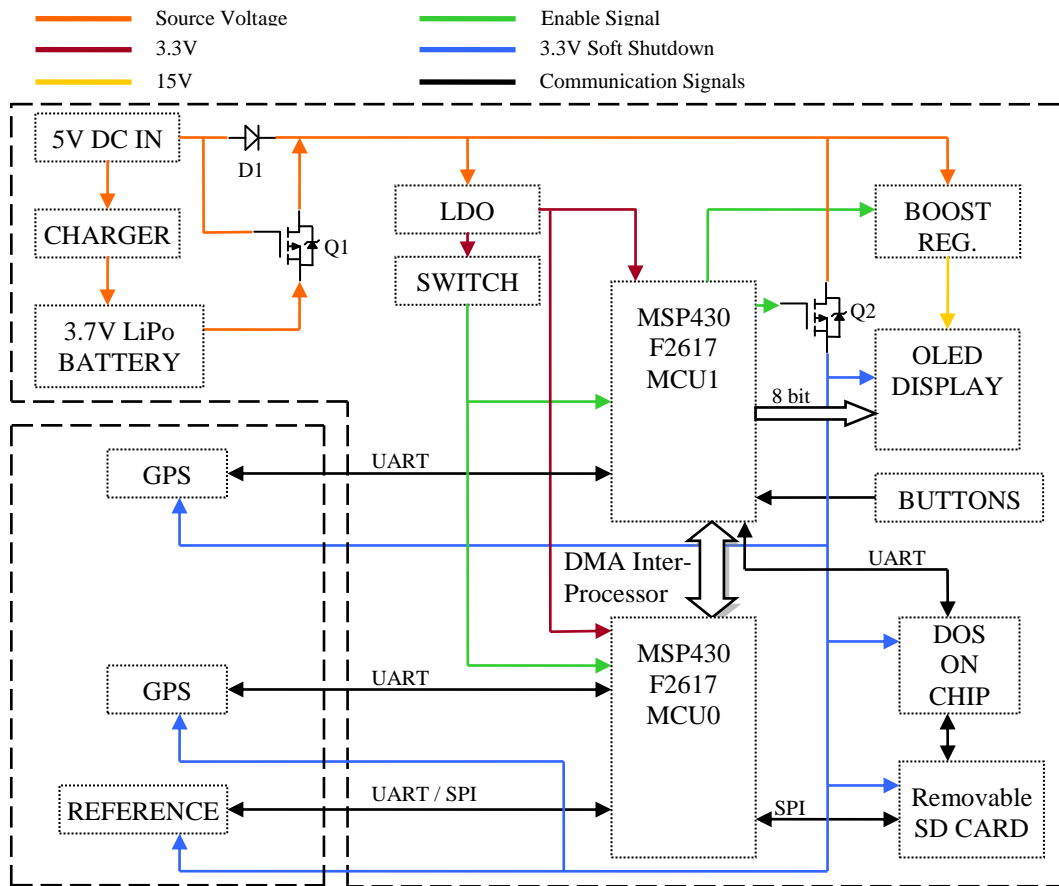


Figure 26: Testbed PCB Revision 2.1 Block Diagram

5.4.1 Power Management

Power for the testbed is provided either by the 3.7 V lithium polymer battery or by an external 5 V source. The two power sources are ORed using diode D1 and a P-

MOSFET Q1. When the external source is connected, Q1 is off and unloads the battery so it can be properly charged. There is some efficiency loss through the diode, but high efficiency is not a concern when using the external power source. Q1 is on when the external power source is not connected, providing a low-loss power path for the battery. This simple power control mechanism is more battery efficient than the traditional two-diode ORing approach.

The power distribution of the testbed is controlled through software as both MCUs have continuous power when a power source is connected. The MCUs remain in Low Power Mode (LPM) 4, the lowest power MSP430 sleep mode, until the power switch is activated. When the power switch is turned on, the MCUs enable power to the remaining circuits. This software controlled power scheme is implemented because it allows for a graceful shutdown of external devices. The devices requiring a graceful shutdown are the GPS modules, the OLED screen, and the SD Card. After the power switch is turned off, the following actions are taken prior to shutdown and entering LPM4:

- A message is sent to the GPS modules to save ephemeris information to non-volatile memory. This reduces startup time for the next power up.
- The OLED screen 15 V power is disabled for at least 100 ms prior to logic power off according to datasheet specification.
- Any data queued to be written to the SD Card is completed.

When the graceful shutdown actions are completed, the power to the rest of the circuit is removed. The MCUs are also able to monitor the battery voltage, and force a graceful shutdown when the battery is nearing the under voltage lockout of 2.8 V. While the testbed is shutdown, 11 μA of current is still consumed by the MCUs and by the LDO quiescent current. If a battery is nearly depleted, it should not remain connected for an extended time of several months or years to avoid slowly over-discharging the battery.

5.4.2 Clock Generation

Both MSP430 processors and peripherals are primarily sourced by the internal DCO oscillator. MCU0 additionally has a 32768 Hz crystal for RTOS tick generation, and to serve as a software Frequency Locked Loop (FLL) base for the DCO. The stable characteristics of the crystal is required for reliable serial port communication as the DCO has considerable frequency drift over wide temperature ranges. The software FLL works by periodically comparing the DCO clock count with the crystal, and adjusting the DCO settings to maintain the desired frequency as closely as possible. The advantage to the software FLL is the capability of having lower power consumption by avoiding a high frequency crystal to drive the UART peripherals. By using the DCO instead of a high frequency crystal, the UART is able to receive characters while the MCU is operating in LPM3 thanks to the very fast startup time of the DCO. If a high speed crystal is used as a clock source for the UART, the MCU cannot operate in a mode lower than LPM0 because of the crystal slow startup characteristics. This would significantly increasing power consumption while the processor is idle.

The clock module of the 2xx series MSP430 has a feature whereby if the configured clock source is stops, it automatically reverts to the internal DCO as the clock source. This feature is exploited by MCU1, as it is configured to be sourced externally from the MCU0 clock output for IPC synchronization. When the transfer is complete and the clock output from MCU0 is disabled, MCU1 continues to operate using the internal DCO.

5.4.3 Memory Storage

The interface to the SD Card removable memory is capable of being connected in two ways for development flexibility, but only one can be used at a time. The first is for MCU0 to communicate directly with the SD Card via a SPI link. In this case, MCU0 must conduct all the low level file structure operations of the File Allocation Table (FAT). The second involves an extra IC called DOSONCHIP. This circuit handles all the low level FAT file operations and writes files to the SD Card,

simplifying development. In this configuration, MCU1 handles the data storage operations. For the first case where MCU0 handles the SD Card communication directly, the DOSONCHIP IC is either not populated on the PCB or is placed in a reset state to avoid contention of the SD Card SPI signals.

5.5 IPC Protocol

The MSP430 does not have an external memory bus, so all communication must happen via one of the serial peripherals, or through the general digital parallel I/O ports for an IPC protocol. The three available serial peripheral choices are the Serial Peripheral Interface (SPI), Universal Asynchronous Receiver Transmitter (UART), or Inter-Integrated Circuit (I²C). SPI has the fastest transfer rate of 4 Mbps (limited by slave device) for a 2xx series MSP430. Rather than using one of the serial peripherals, the digital parallel IO ports with a Direct Memory Access (DMA) controller is explored for a faster and more energy efficient protocol [27]. This protocol, called the Parallel Inter-Processor Communication (PIPC), is applicable to the general case of multiple processors, although the embedded system for this thesis uses only two processors. For the case where more than two processors are used, the PIPC protocol allows for a processor to send a data packet as one-to-one, one-to-many, or one-to-all types of transfers. This capability is highly beneficial for broadcast messages.

Using a parallel data bus goes against the trend of modern computer systems, where a migration has happened from parallel to serial buses for hard drives and graphics cards. For such systems, maintaining signal integrity becomes increasingly difficult beyond about 100 MHz. For the frequency domain of low power embedded systems however, signal integrity is generally not a constraint for a parallel bus with signal lengths less than 10 cm.

5.5.1 PIPC Protocol Operation

For the MSP430 microcontroller, digital IO normally requires 6 clock cycles for read/write operations of the port. By contrast, the DMA controller can read/write

data from/to the ports every two clock cycles, so it is highly advantageous to make use of the DMA controller. When two 8-bit ports are combined on the MSP430 to make a 16-bit data bus, two bytes are transferred per cycle. Thus for a 16 MHz processor, a maximum throughput of 16 MB/s is possible.

To ensure successful data transfer, the read/write operations need a

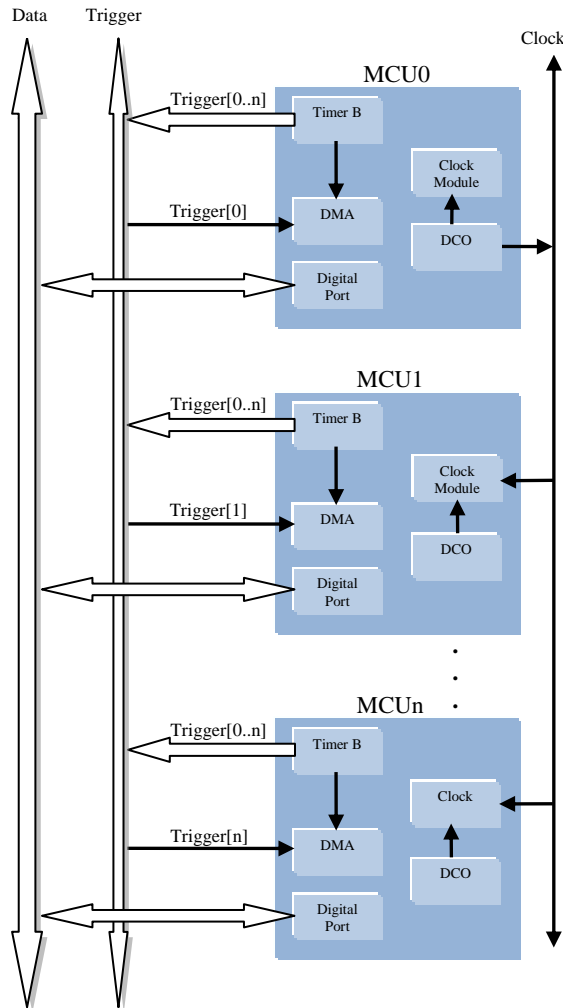


Figure 27: PIPC Synchronization Signals

synchronization mechanism between the sending and receiving MCU(s), both for initiating the transfer and maintaining synchronization. This is accomplished by using the timer B peripheral to initiate the transfer, and by using a common clock to maintain read/write synchronization, as shown in Figure 27.

The clock is provided by MCU0 to all other MCUs for the duration of the transfer. The gating of the clock signal is discussed later in the chapter. Each MCU is configured to use the external clock signal by default but reverts to using the internal DCO when there is no PIPC transfer occurring. The transition of using the external clock sourced from MCU0, or the internal DCO happens automatically

within the clock module. This clock scheme allows each MCU to operate in any low power state and still be capable of receiving a PIPC transfer at any time.

The sending MCU uses a DMA channel to write data from a block of memory to a single output port. The receiving MCU uses two DMA channels to read data from

a single input port and fill a block of memory. The transfer is initiated by a trigger signal going to both the sending and receiving MCU DMA controllers. The sending DMA controller is configured to acquire a trigger signal internally from the Timer B peripheral, while the receiving MCU receives the trigger signal via a hardware pin that is also controlled by the Timer B Output Unit. The Timer B peripheral is set up with the following configuration:

- Timer B is set to Up Mode.
- Capture/Compare Register 0 (CCR0) is set to the value of three. CCR0 triggers the sending DMA controller internally.
- Capture/Compare Registers 1 to 6 (CCR1-6) are set to the value of one. CCR1-6 triggers the receiving DMA controller via an output pin controlled by the Timer B OUT signal of the sending MCU. The OUT signal is configured for Set Mode 001, which asserts the pin when the CCR register value is reached in the Timer B counter. The protocol software driver enables only the pins for the intended destination MCUs for each transfer instance.

The receiving MCU has two DMA peripheral channels involved in the receive operation. CH0 is configured to receive a single word on the data bus and transfer it to the size configuration register of another separate DMA channel, CH1. The transfer size word is placed on the data bus by the sending MCU before initiating the trigger. Both CH0 and CH1 receive channels are triggered simultaneously via the hardware pin, but CH0 has a higher priority and activates first, readying CH1 with the block size of the ensuing block transfer. CH1 begins immediately after the CH0 is complete and transfers the number of words configured by CH0. Figure 28 shows the PIPC trigger timing diagram.

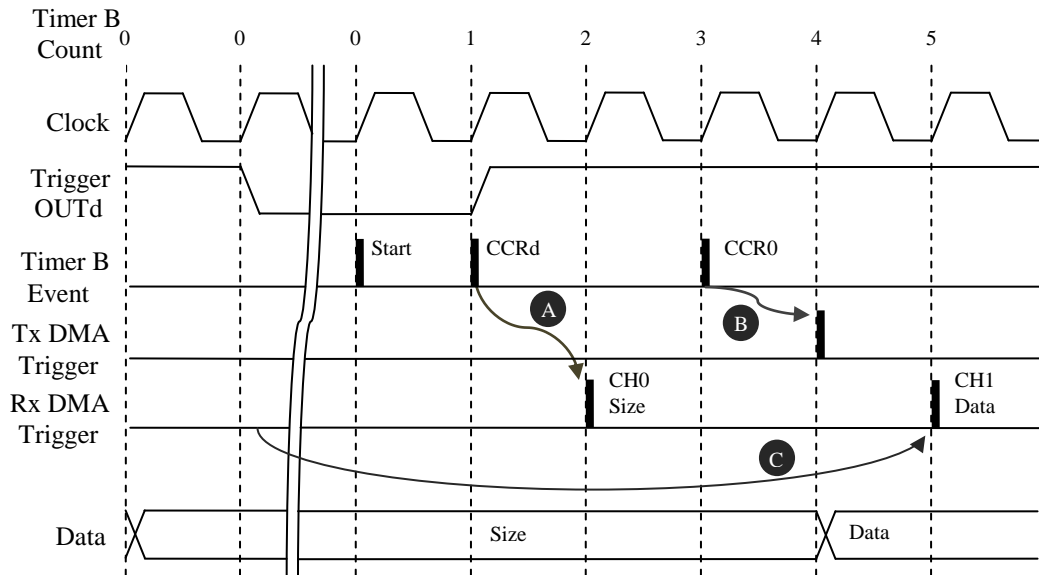


Figure 28: PIPC Protocol Trigger Timing Diagram

The

PIPC transfer sequence is as follows:

1. Transfer size is placed on the data bus by sending MCU
2. The trigger output of the intended receiving MCU(s) is de-asserted. Subscript d denotes the receiving MCU ID(s).
3. Timer B counter is started.
4. Trigger output(s) d are asserted automatically by Timer B of sending MCU when CCRd count of one is reached. This causes the receiving MCU(s) DMA CH0 to be triggered, denoted by marker A.
5. DMA CH0 copies data size from data bus to size register of CH1 of receiving MCU(s) d .
6. The DMA of Sending MCU is triggered internally by CCR0 when Timer B counter reaches 3, denoted by marker B.
7. Receiving MCU(s) d begin transferring data from bus to block of memory when CH0 is complete, denoted by marker C. The CH0 transfer takes two clock cycles, plus an additional wait cycle is needed by the DMA controller before CH1 begins.

Figure 29 shows how once the transfer is initiated, each data word remains on the bus for two clock cycles with the sending write cycles interleaving the receiving read cycles.

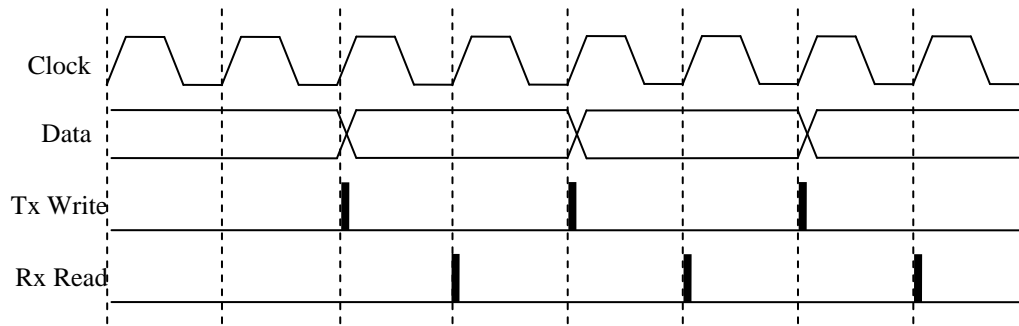


Figure 29: PIPC Protocol Data Transfer Cycles

5.5.2 PIPC Protocol Media Access

Whenever multiple entities are using a common bus system, there must be a mechanism for sharing the bus to avoid signals contentions. In other words, there must never be more than one MCU attempting to drive the data lines at any time. All MCUs must place the data port into a high impedance state unless there is exclusive access to the data lines. This is accomplished in the PIPC protocol via arbitration. MCU0 is the arbiter for the PIPC protocol as it is responsible for handling the request and grant signals. Whenever a MCU wants to send data to another MCU, a bus request is first made to the arbiter. When the arbiter determines the bus is free to be used, a bus grant assertion is given back to the requesting MCU. The requesting MCU is now the bus owner and may drive the data lines. Once the data transfer is complete, the requesting MCU releases its request line, signaling to the arbiter that it is no longer the bus owner. The arbiter is now free to grant the bus to another requesting MCU. For the case when there are more than one MCU requesting the bus simultaneously, the arbiter uses a priority based scheme, where the lowest numbered MCU has the highest priority. A lower priority MCU that is requesting the bus will not be given access to the bus until all higher priority MCUs have been given prior access.

In addition to the bus request and grant signals, each MCU also has busy output signal. This signal is used to prevent buffer overruns. This busy signal of the receiving MCU must be checked by the sending MCU prior to initiating a data transfer. Figure 30 shows the arbitration signals of the PIPC protocol.

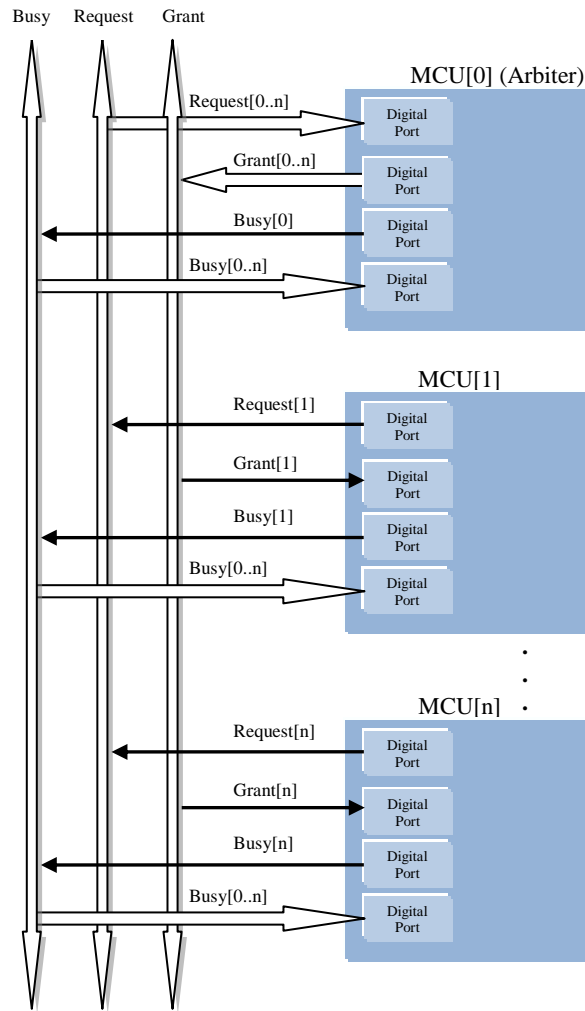


Figure 30: PIPC Arbitration Signals

The arbitration sequence in Figure 31 is as follows:

1. The sending MCU asserts the Bus Request line.
2. When the bus is available, the arbiter asserts the Bus Grant line and provides the external synchronization clock.
3. The sending MCU initiates the synchronized DMA transfer.
4. As soon as the transfer is complete, the sending MCU de-asserts the Bus Request line.
5. When the Arbiter senses the Bus Request line has been de-asserted, the Bus Grant line is de-asserted and the clock output is disabled.
6. The receiving MCU asserts its Busy line until the receiving buffer is cleared.

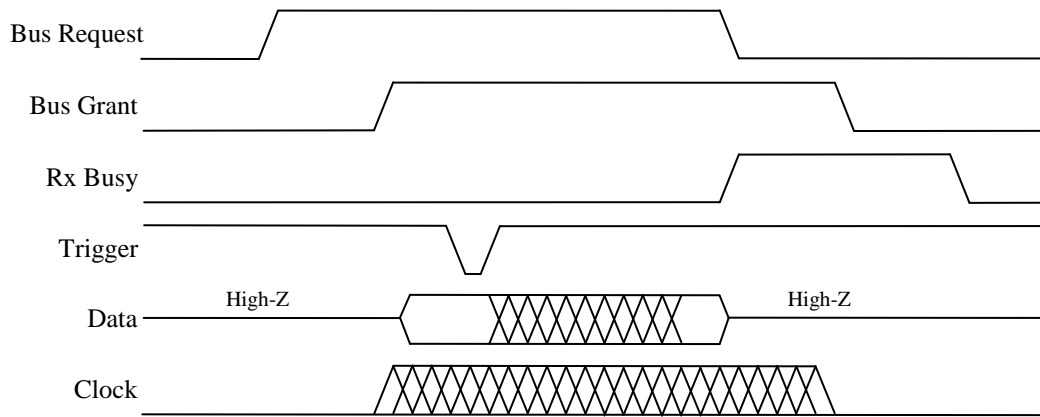


Figure 31: PIPC Arbitration Timing Diagram

5.5.3 SPI Protocol Operation

In comparison to the PIPC protocol, an alternate IPC system based on the SPI protocol with a media access scheme identical to the PIPC protocol is also explored. This protocol will be referred to as the SPIIPC. The media access control is not typically required for the SPI protocol with a single master and single/multiple slaves. However, it is needed for the case of two or more MCUs where each MCU has the capability of assuming either the master or slave role depending on the configuration of each individual data transfer. Each MCU is configured to be a slave device by

default, unless it has obtained control of the bus from the arbiter and becomes a master for a data transfer. Once one of the MCUs has control of the bus, the system SPIIPC behaves the same as a typical SPI protocol. To make the data transfer as efficient as possible in terms of throughput and current consumption, DMA controllers are used for both the sending and receiving MCUs. No external signals are required to trigger the DMA controllers, as the SPI peripheral provides internal triggers directly to the DMA controller. Figure 32 shows the connections for the SPIIPC signals.

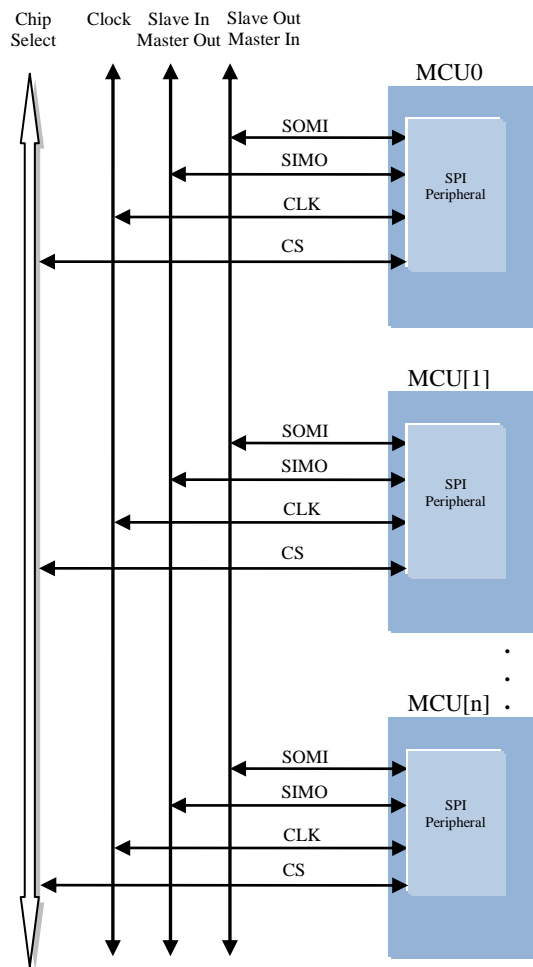


Figure 32: SPIIPC Inter-Processor Signals

5.5.4 Inter-Processor Communication Performance Measurements

The maximum throughput of the SPIIPC and PIPC protocols is compared in Figure 33 for various MPU clock speeds. The PIPC protocol has the clear advantage, with a maximum throughput of 16 MB/s for a 16-Bit data bus which scales linearly up to the maximum clock speed. The SPIIPC protocol has a maximum throughput of 0.5 MB/s which is reached at a MPU clock speed of 4 MHz. MPU clock speeds beyond 4 MHz have no effect on SPIIPC throughput as it is constrained by the MSP430 slave peripheral electrical timing specifications.

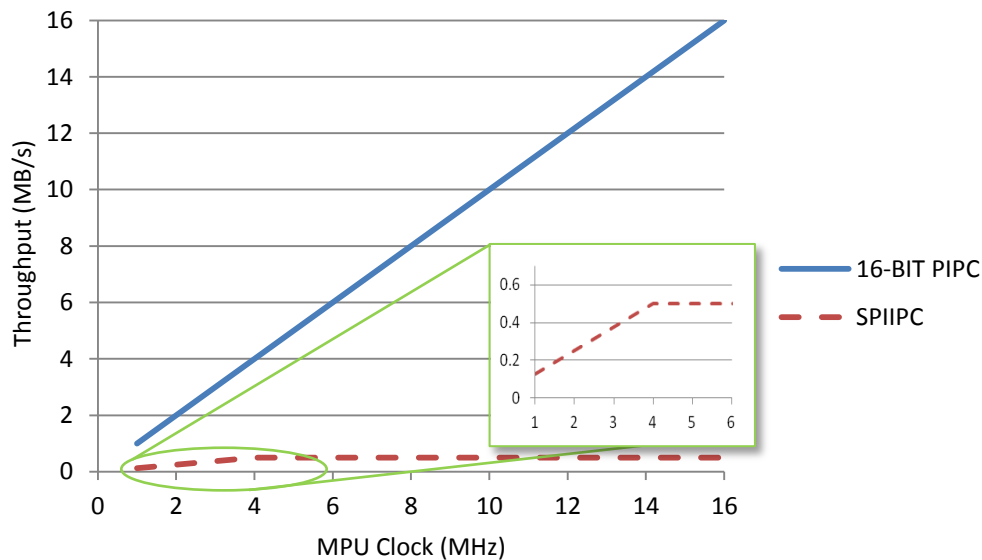


Figure 33: IPC Protocol Maximum Throughput

The maximum throughput in Figure 33 is not actually achievable in practice because of the overhead associated with media access. Media access is the same for both protocols, and was measured to take 106 clock cycles from the time the bus request is made until data transfer commenced. The measurement was performed with an oscilloscope on over 1000 consecutive data transfers with the same result. The media access is not deterministic however, and there are conditions that may cause longer access times. A bus request while the arbiter temporarily has interrupts

disabled would prolong gaining access to the bus. A sending MCU may also encounter a busy signal on an intended receiving MCU and be required to wait before requesting access to the bus. The effective throughput was calculated based on the maximum transfer speed and media access overhead for various data packet sizes for both protocols. The maximum MPU clock speed of 16 MHz was used for the calculations. The results are shown in Figure 34.

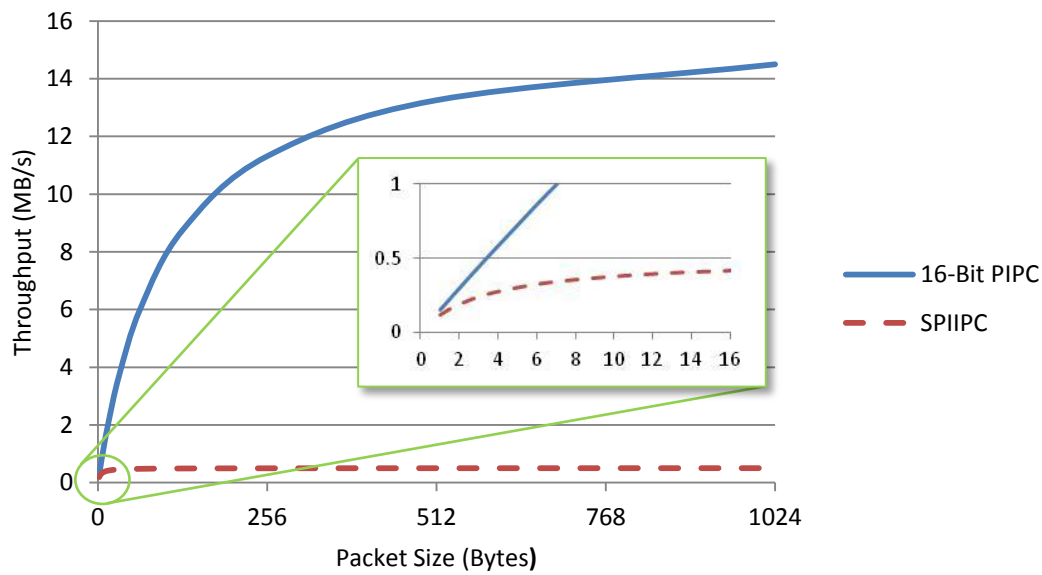


Figure 34: IPC Protocol Effective Throughput

Both protocols asymptotically approach the maximum throughput as the packet size increases, with the SPIIPC reaching 95% of its maximum throughput at 64 bytes and PIPC reaching 95% of maximum at a packet size of 2 kB. The media access overhead is not significant for the SPIIPC protocol because of the relatively slow transfer rates. Conversely, lower packet sizes do significantly affect the throughput of the PIPC protocol, but it is still far superior to SPIIPC whenever the packet size is larger than two Bytes, and has an order of magnitude better performance for packet sizes larger than 32 Bytes.

The energy efficiency is also compared for the PIPC and SPIIPC protocols. The total current consumption of two MCUs involved in a IPC data transfer was measured with all other PCB ICs and inputs disabled or disconnected. For the PIPC protocol, both the sending and receiving MPUs DMA controllers are continuously active for the duration of the transfer. For the SPIIPC protocol, the sending MCU operates in LPM0 with the DMA controller intermittently active as bytes are sent, and the receiving MCU is in LPM3, with the DMA controller intermittently active as bytes are received. The data pattern for the SPIIPC protocol was set up with repeating '0x55' bytes, while the data pattern for the PIPC protocol was set up with alternating '0x5555' and '0xAAAA' words in order to maintain the highest possible odd-mode switching currents. As seen in Figure 35, the current of both IPC protocols increase linearly with throughput.

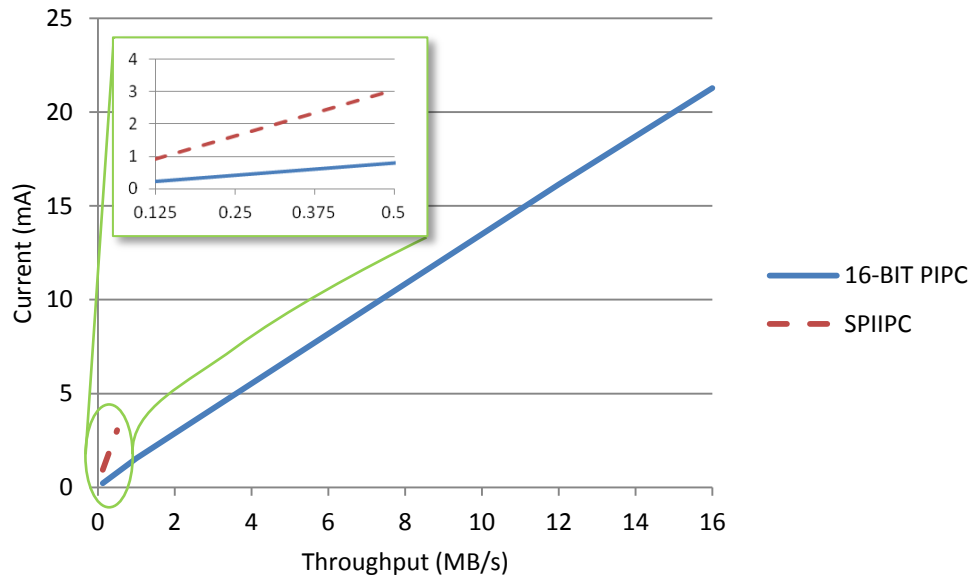


Figure 35: IPC Protocol Current Consumption

Figure 36 shows the energy efficiency in terms of the transfer rate ratio. Here it is clear the current efficiency of the PIPC protocol is significantly better than SPIIPC, with the SPIIPC protocol consuming approximately three times the current as the PIPC protocol.

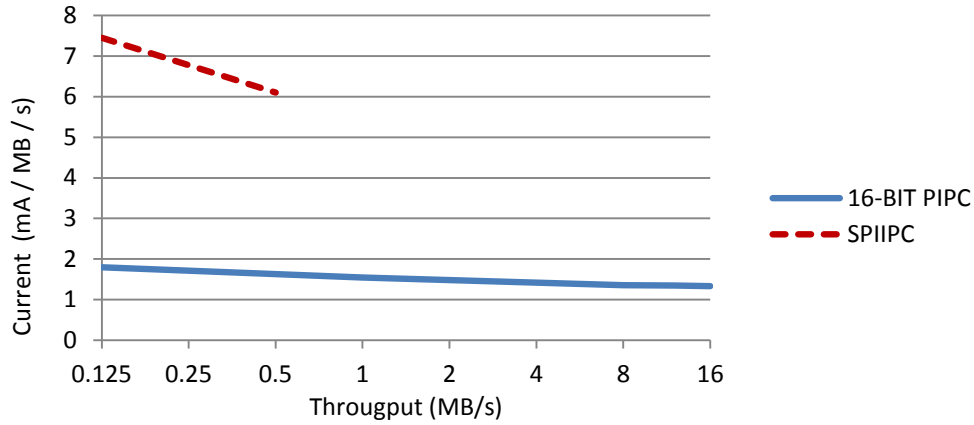


Figure 36: IPC Protocol Current Ratio

The line switching current of the protocols was also measured. The measurements were performed by measuring the current consumption of a single MPU during the protocol transfer with and without the output pins enabled. The line switching consumption is the difference between the two measurements. Figure 37 shows the switching current ratio at varying transfer speeds.

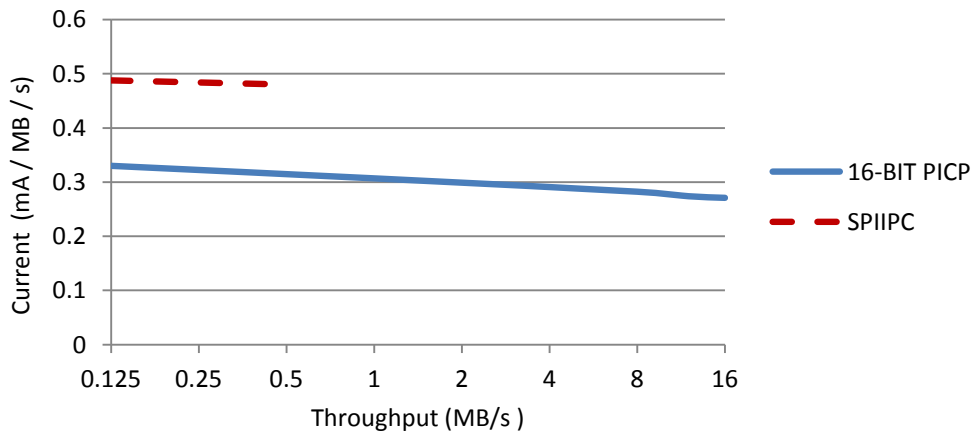


Figure 37: IPC Protocol Switching Current Ratio

The PICP protocol has a slight efficiency advantage regarding switching currents. This is as expected because for the SPIIPC protocol, the synchronous clock has a transition for every bit of data transferred, while the PIPC protocol has a synchronous clock transition for every 16 bits transferred. The communication physical trace lengths are on the order of 5 cm on the testbed hardware. A system with many processors with much longer trace lengths could significantly add to the switching current figures. Even so, the switching current is small compared to the overall current consumed by the processors during the transfer.

The PIPC protocol is superior to SPIIPC in regards to speed performance and energy consumption, both which are important design considerations. However, there are drawbacks to the PICP protocol. It has a higher pin count than SPIIPC, and requires the use of the Timer B peripheral. The clock configuration is also constrained, with less flexibility than SPIIPC as all MCUs must operate at the same frequency.

5.5.5 Inter-Processor Communication Pin Requirements

The pin count of both protocols is compared in Table 9. For both protocols, the media access control pin requirements are the same. There are $3N$ signals required where N is the number of processors. For PIPC, $1N$ trigger signals are needed, 16 signals for data, and one for clock. For SPIIPC, 3 SPI (CLK, SIMO, SOMI) and $1N$ Chip Select signals are required.

Table 9: IPC Protocol Pin Requirements

Protocol	Media Access	Data	Trigger / CS	Clock	Total
PIPC	$3N$	16	$1N$	1	$4N + 17$
SPIICP	$3N$	2	$1N$	1	$4N + 3$

5.5.6 Inter-Processor Communication Applications

The PIPC protocol has applications beyond the testbed developed for this thesis. The use of parallel processing is especially suited for wireless sensor networks because a sensor node is in an idle state for the vast majority of the time, with a need

for high performance processing only when some sporadic event happens. The requirement for extremely low power consumption during idle sleep modes and also high performance capabilities exists for many applications. The difficulty is that there are few processors that meet both criteria. Many high performance processors have relatively high idle power consumption, while many low power processors do not have exceptional processing capabilities. An approach to the problem is to use multiple low power processors in a performance scalable architecture. The idle power of the multiple MSP430 MPUs combined could be lower than a high performance processor, and high performance computing is achievable if the application is able to make use of parallelism. The PIPC protocol allows for highly efficient and high performance inter-processor protocol that minimizes communication overhead and maximizes the achievable performance of a multi-processor design.

5.6 Testbed Power Measurements

Table 10 shows the current consumption of the testbed PCB with individual components enabled or disabled. When no input devices are connected and the power switch is off, the PCB consumes 11 μA of quiescent current. When the power switch is turned on, the two MCUs and all fixed PCB components consume 533 μA of idle current. This value is with the OLED display removed. When the OLED display is connected, it consumes an additional quiescent current of 547 μA , and 27.5 mA when switched on. Attaching a Trimble Copernicus GPS module adds 34.7 mA. Inserting a SD Card adds 19.7 mA when the system is logging input data.

Table 10: Testbed Current Consumption

Mode	Current (mA)
Off	0.011
Idle	0.533
GPS	34.7
SD Logging	19.7
OLED IC I _q	0.547
OLED ON	27.5

5.7 Testbed Hardware Summary

Three PCB revisions were designed and built during the development of the ADS testbed hardware. The progression of the design included improvements to the clock system, IPC design, power management, storage and better flexibility of GPS input devices. Other design changes intended to be improvements were not successful, such as upgrading the MPU to the newest available hardware. Software tool compatibility issues forced abandonment of the MPU design changes and revert to a previous design iteration.

A new IPC protocol was designed and successfully implemented for multi-processor systems. Performance measurements of the PIPC protocol demonstrated it is up to 32 times faster and consume three times less current than the alternate SPI-based IPC protocol. This new protocol allows for scalable multi-processor applications that are not hindered by large overhead that could negate any performance improvements of a multi-processor over a single processor design.

Chapter 6: Software Development

6.1 Software Development Introduction

At the onset of software development, Imagecraft's ICCV7430 was one of the only Integrated Development Environments (IDE) capable of programming the 16 MHz 26xx series MSP430 MPUs. In addition, ICCV7430 included a binary library for the Tiny version 3.2.3 of Pumpkin Salvo RTOS. For these reasons, the Imagecraft IDE tools were initially chosen for software development. A change of software tools was made near the completion of software development for compatibility reasons discussed in Chapter 5. Additionally the Imagecraft ICCV7430 tools became categorized as End-of-life without license renewal, necessitating alternate tools. The Texas Instruments Code Composer Studio v5 (CCSv5) was chosen as a replacement. By the time the new CCSv5 IDE tools were chosen, it was fully compatible with the MSP430 26xx series and included a full-featured RTOS called SYSBIOS.

6.2 Real-Time Operating System Overview

An RTOS provides scheduling algorithms in order to guarantee process deadlines will be met, with a process being a collection of code, memory, and data in an execution environment. All RTOSs have a *kernel*, which is the part of RTOS that is responsible for thread management, inter-thread communication and synchronization. A thread is the basic sequence of code to which the RTOS allocates processor time within the scope of the process. The Scheduler is the part of the RTOS kernel that is responsible for deciding which thread should be executing at any given time.

6.3 SYSBIOS Real-Time Operating System

The SYSBIOS RTOS is designed to be used by applications that require real-time scheduling and synchronization. Some features of SYSBIOS include preemptive multi-threading, hardware abstraction, and tick suppression. SYSBIOS is designed to minimize memory and processing requirements on the CPU target.

SYSBIOS provides support for several types of program threads with different priorities. Each thread type has different execution and preemption characteristics. The thread types, from highest to lowest priority are:

- Hardware interrupt (HWI) threads. HWIs are the threads with the highest priority in a SYSBIOS application and are the equivalent to Interrupt Service Routines (ISRs). HWI threads are used to perform time critical tasks that are subject to hard deadlines. They are triggered in response to external asynchronous events (interrupts). HWI threads cannot normally be preempted on the MSP430, and thus run to completion once invoked even if a higher priority HWI is pending.
- Software interrupt (SWI) threads. SWI threads provide additional priority levels between HWI threads and Task threads. SWIs are similar to HWIs, but are triggered by software calls as opposed to HWIs which are triggered by hardware interrupts. SWIs are useful for threads that are subject to tight time constraints, but whose deadlines are not as firm as those of HWIs. SWIs allow HWIs to defer less critical processing to a lower-priority thread. This minimizes the time the processor spends inside a HWI thread, where preemption by all other HWIs are disabled. SWIs may be preempted by another SWI thread with higher priority, or by any HWI.
- Task threads. Task threads have higher priority than the background (Idle) thread and lower priority than SWIs. Tasks differ from SWIs in that they can wait (block) during execution until necessary resources are available. Active tasks exist in one of three states: running, ready, or blocked. Only one task can be running at any time, as it is the task being executed. A task that is in the ready state will execute when the scheduler determines it is the highest priority task. A task that is in the

blocked state cannot execute until a particular event occurs within the system such as a resource becoming free or a timer expiring. SYSBIOS provides a number of mechanisms that can be used for inter-task communication and synchronization. These include Semaphores, Events, Message queues, and Mailboxes. Tasks may be preempted by another task thread with higher priority, or by any SWI or HWI.

- Idle thread. Idle threads execute at the lowest priority in a SYSBIOS application and are executed one after another in a continuous loop. The Idle Loop runs continuously except when it is preempted by any other thread. For the MPS430 processor, the idle thread usually consists of operating in one of the low power modes where the processor is halted.

6.4 Testbed Programs

The testbed programs are comprised of various tasks, HWIs, SWIs and message queues as depicted in Figure 38 and Figure 39. The SYSBIOS kernel scheduler determines which task should run based on each task's state and priority. Data is passed from one task to another via events and circular buffer message queues.

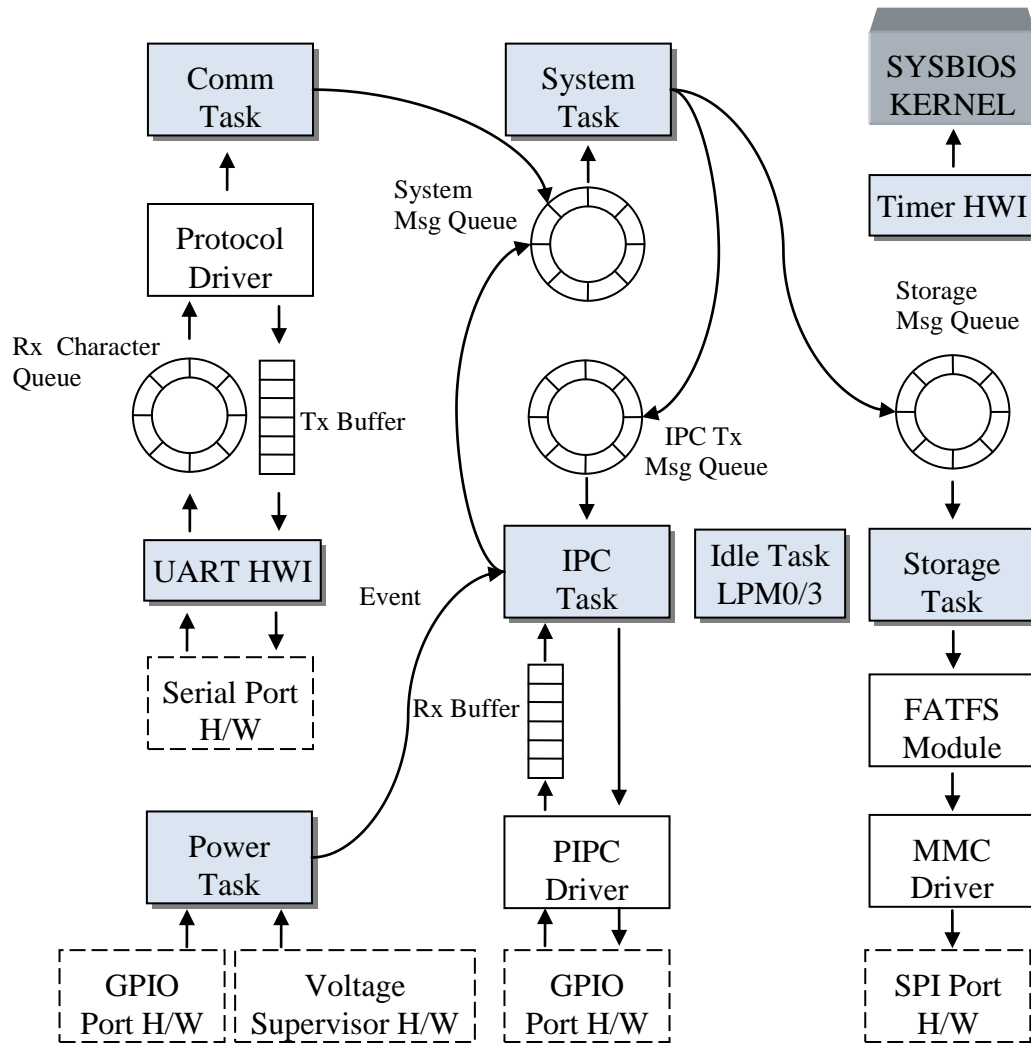


Figure 38: Testbed MCU0 Program Data Flow

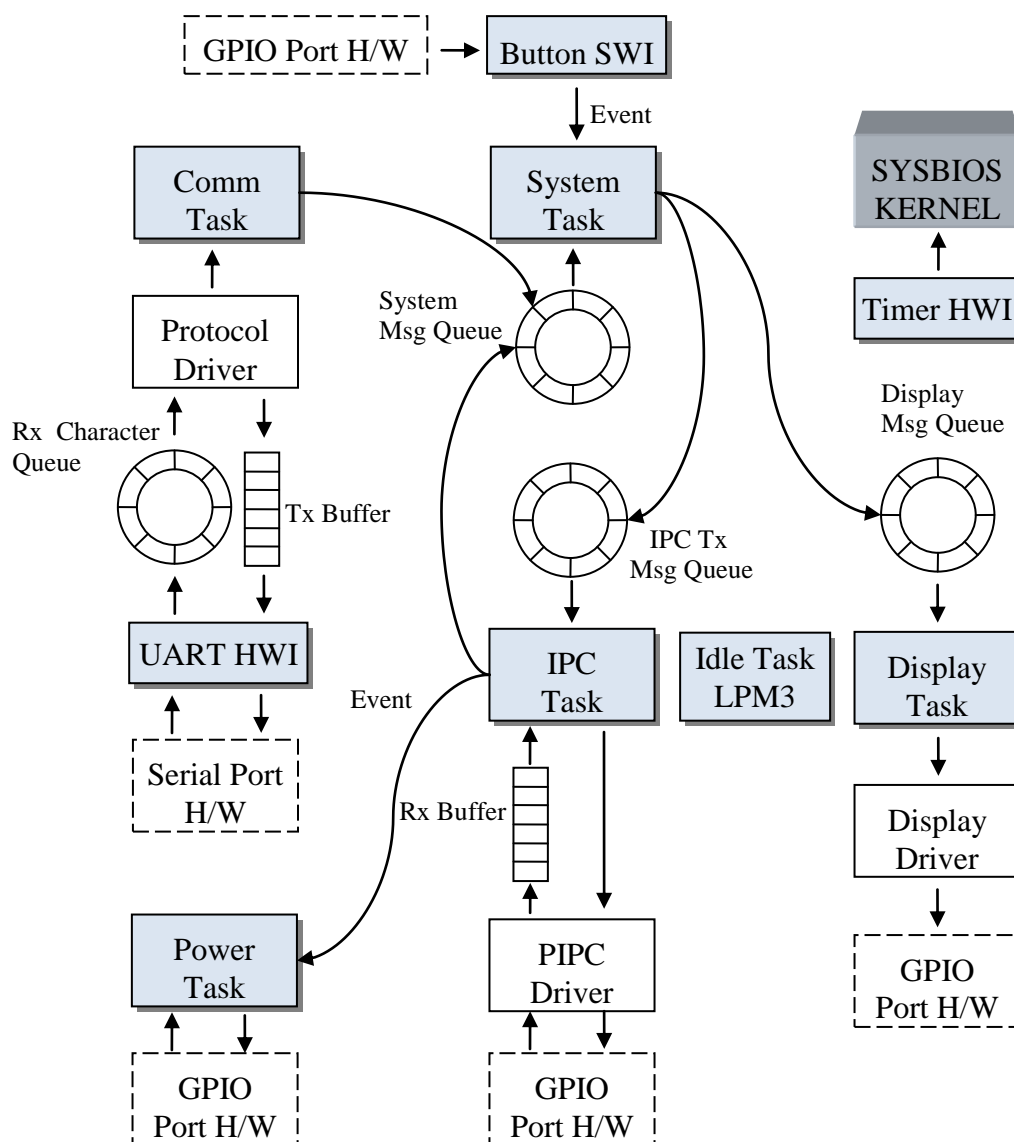


Figure 39: Testbed MCU1 Program Data Flow

The circular buffer message queues follow a producer - consumer model. There may be more than one producer for each queue, but only one consumer. A producer task requests a memory block from the queue heap memory. After the memory block is allocated, the producer task places a message on the queue, then sends an event to the consumer task of the queue. The consumer task of each queue

processes messages as execution time is allocated by the kernel. Once the message is processed, the memory is released back to the queue to be reused.

6.4.1 Program Tasks

The Comm Tasks of both MCUs is the same. The purpose of the Comm task is to gather data packets from the attached hardware and present messages to the System Task in a unified way, regardless of the type of communication protocol or type of hardware is attached to the serial port.

The IPC Task is also the same for both MCUs. The IPC Task is responsible for exchanging data between MCUs. Data received by the task from the opposing MCU is presented to the System Task, and data sent to the task will be delivered to the opposing MCU's IPC Task.

The Storage Task exists only on MCU0. It receives messages from the System Task and uses the FATFS routines to write logs to the SD Card.

The Display Task exists only on MCU1. Display driver routines are used to create text or graphics based on messages received from the System Task.

For the System Task, significant differences exist between each MCU. The only similarity is that for both MCUs the System Task gathers data from the Comm and IPC Tasks and also sends messages to the IPC Task. For MCU0, the System Task delivers messages to the Storage Task. This includes data received from the opposing MCU via the IPC Task. Data acquired locally from the Comm Task is also sent to the opposing MCU via the IPC Task. For MCU1, the System Task consists of a state machine representing the operating mode testbed. Based on the operating mode, calculations are performed, data is formatted and messages are sent to the Display Task. Messages from the local Comm Task or the opposing MCU via the IPC Task are processed and formatted for the Display Task.

The Power Task for MCU0 monitors the power switch and supply voltage. A change in the power switch state or a low voltage condition will trigger an event to be sent to the IPC Task to alert MCU1. The Power Task of MCU1 receives the event,

and performs sequenced power up/down actions to the rest of the circuit. Circuits requiring specific power sequencing include the GPS receivers, OLED display and SD Card.

The idle task is managed by the SYSBIOS configuration and simply enters a LPM until a wakeup condition occurs. MCU1 is able to use LPM3 in the idle task, while MCU0 uses either LPM0 or LPM3 depending on the state of the PIPC arbitration driver. If MCU0 is providing the external clock for an IPC transfer, then LPM0 is the lowest mode available. Otherwise, LPM3 is used in the idle task.

6.4.2 Program HWIs

The Timer HWI in both MCUs generates 10 ms period tick events for SYSBIOS. These events are used by SYSBIOS for sleep, timer and timeout features of the RTOS.

The UART HWI is simply the ISR for sending and receiving single characters via the peripheral and the data buffers. The protocol drivers take care of moving larger data segments on and off the buffers.

6.4.3 Program SWIs

The Button SWI is called periodically to poll the state of the user buttons. Normally a HWI would be used to detect a button press, but hardware ports capable of interrupts were already used by other circuits in the design. A detected button press results in an event being sent to the System Task state machine.

6.5 File Formats

The testbed stores information to a removable media SD Card. The data is saved in a FAT file format to be accessible on a PC. The files are stored under the GPSLOG folder of the SD Card. The naming convention for the files is as follows: `wwwmmmm.gps`, where *w* is the 4-digit GPS week number, and *m* is the minute of the week of when the file was created.

The *.GPS file packets consist of the GPS time, then the elevation, azimuth, and SNR level for each of the 32 GPS SV PRNs as shown in Figure 40. The packets are wrapped with header and footer words for delineation. The brackets indicate the number of bytes for each field.

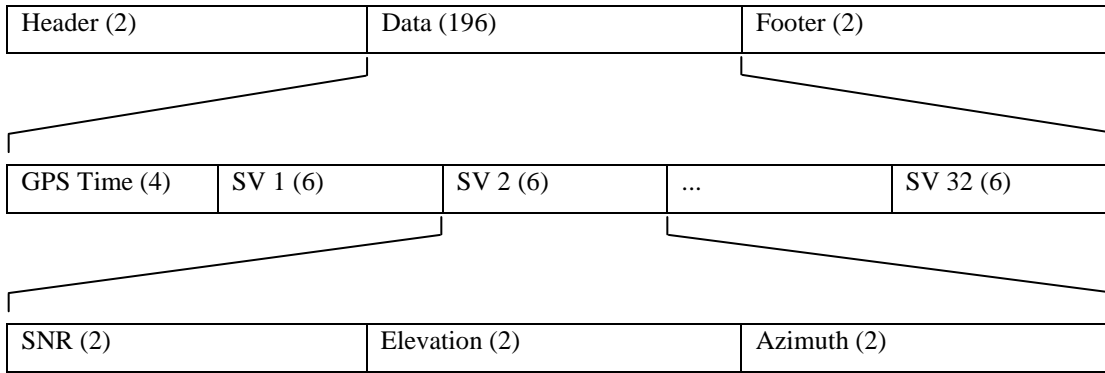


Figure 40: GPS File Formatting

The SNR, elevation and Azimuth initially are all floating point numbers on the testbed before they are written to the SD Card. To save space, they are converted to fixed point integers by multiplying the values according to Table 11. When the files are later processed, the numbers need to be divided by the correct values and converted back to floating point data types.

Table 11: GPS File Data Types

Field	Date Type	Description
Header	Integer	0x5555
GPS Time	Long	Time in seconds since start of current GPS week
SNR	Integer	= $\text{SNR} \times 10$ (dB-Hz or AMU)
Elevation	Integer	= $\text{Elevation} \times 1000$ (radians)
Azimuth	Integer	= $\text{Azimuth} \times 1000$ (radians)
Footer	Integer	0xAAAA

6.6 Testbed Operation

The testbed is turned on and off by an illuminated switch adjacent to the power connector. The switch is illuminated orange when external power is connected and the battery is charging. Once the battery charge is complete, the switch illuminates green. The OLED screen also has a power switch so that it may be powered on to configure the testbed, then powered off to conserve power while data logging. Whenever the screen is powered by the boosted 15V supply, the OLED power switch is illuminated in red.

Whenever a SD Card is present during power up, the testbed will begin logging data as soon as GPS module packets become available. The data logging occurs regardless of the operating mode of the testbed to a file with the naming convention described previously in this chapter.

There are five modes of operation on the testbed, shown in Figure 41, as it appears on the OLED display main menu. The UP / DOWN buttons highlight the various modes, and is indicated when the mode illuminates white. The SELECT button chooses the highlighted mode. The MENU button returns the screen to the main menu from whatever mode the testbed had active.

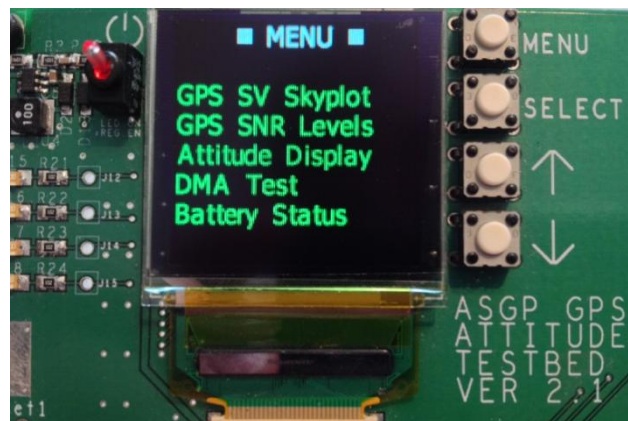


Figure 41: Testbed Menu

6.6.1 GPS SV Skyplot

The skyplot mode displays graphical information about the SV constellation and the signal strength of each tracked SV. The skyplot is represented by a polar coordinate plot, where the outer periphery of the circle represents the horizon, and the center of the plot represents the zenith direction. Concentric circles from the center show 60, 30, and 0 degrees elevation. The azimuth is labeled in 45 degree increments, with zero being North, and ninety being East. When a SV orbit is known by the GPS, the SV PRN number appears in green. When the SV signal is acquired, the SV number appears in red, accompanied by a signal strength bar above the number.

6.6.2 GPS SNR Levels

The SNR Levels mode shows a bar graph of up to eight SV signal levels. A peak marker also shows the maximum SNR value used for normalization. When two GPS receiver modules are connected to the testbed, the SNR values are shown as red and blue bar graphs to differentiate the modules.

6.6.3 Attitude Display

The attitude display shows the calculated attitude reference using the same polar plot as the SV Skyplot mode. The attitude vector is displayed by a dot representing the endpoint of a unit vector. The attitude is only displayed if three or more SV signals are available.

6.6.4 DMA Transfers

The DMA Transfer mode displays packet information about the inter-MCU high speed communication protocol transfers. It is a testing mode to validate timings and reliability of the data transfers, including any transfers with errors or lost packets.

6.6.5 Battery Status

The Lithium Polymer cell voltage is displayed in the battery status mode. An estimated battery percentage is also displayed. The battery life is calculated using a linear function, and is less accurate when the cell is fully charged or nearly discharged.

6.7 Software Development Summary

Software development began with the Imagecraft ICC development tools, but a migration to the Texas Instruments Code Composer Studio tools was made late in the development stage because of compatibility and support issues. This change also involved changing the RTOS from Pumpkin Salvo to Texas Instruments SYSBIOS.

The final software implementation has a common template for both MCUs in terms of how the tasks are structured. The tasks use RTOS synchronization mechanisms and message queues for inter-task communication. The tasks are compartmentalized to more easily manage the complexity of the overall program. The software is modular, where additional tasks can be added without requiring alterations to the existing parts of the program.

Chapter 7: Test Setup and Results

7.1 Test Setup

The measurement truth reference of the SNR ADS was a Septentrio PolaRx2 carrier phase GPS receiver, configured with two antennas separated by a 1 m baseline. The two Septentrio reference antennas were fixed to the ends of a wooden base atop a tripod, while the experimental SAP antenna was mounted in the middle of the two reference antennas. For this configuration, the reference attitude accuracy is 0.3° for heading and 0.6° pitch [28]. Figure 42 illustrates the test setup.

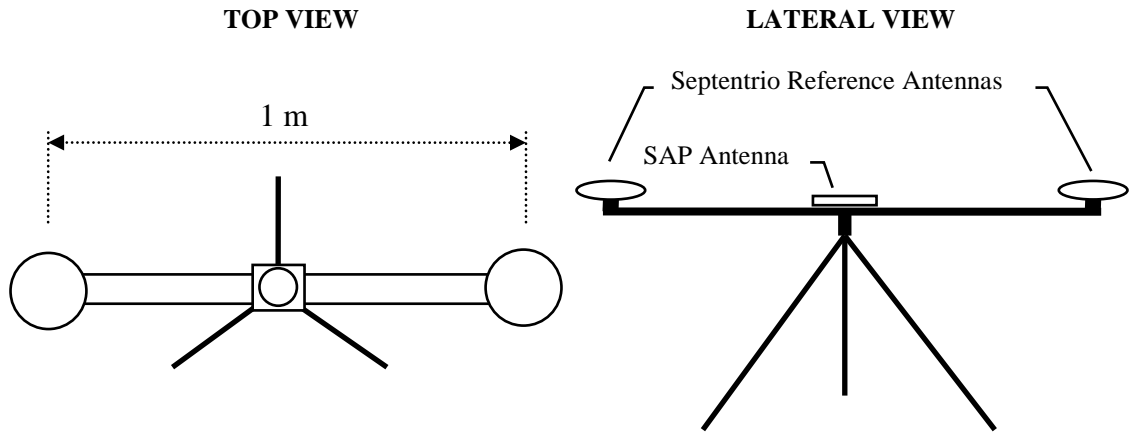


Figure 42: Measurement Test Apparatus Setup

The Septentrio was used as a reference for all non-zenith pointing measurement sessions. For the zenith pointing sessions, a spirit level with 0.5° accuracy was used.

7.2 Test Measurement Sessions

Five measurement sessions were performed in the Fairbanks area and analyzed for accuracy. The first session was conducted at Creamers Field where the SAP antenna was set to a zenith attitude. The second session was on an urban rooftop, pointing 19° off-zenith in a North-Westerly direction. The third was conducted at the top of Ester Dome, with the SAP antenna pointed at 45° off-zenith in a South-Westerly direction. The forth session was conducted near Hez Ray baseball field, again

configured to a zenith attitude. The fifth session was also at Hez Ray Field, but with the attitude set to 22.5° off-zenith in a southerly direction. Most of the measurement session had an un-obscured view of the sky apart from terrain up to about five degrees above the horizon. The exception was the Urban Rooftop session which had considerable pine tree obstructions of the sky. The five static measurement session details are shown in Table 12. The sessions pointing in the zenith direction generally had more SV signals within view compared to those sessions where the attitude was more off-zenith. The reference attitude vector is defined with the positive \hat{x} direction pointing North, and the positive \hat{y} direction pointing West, and the positive \hat{z} direction pointing up. The data collected during the measurement sessions was stored on a SD Card as described in Chapter 5. The analysis was then performed using MATLAB to open the files, compute the attitude solution, and graph the data.

Table 12: Measurement Sessions

Location	Date	Duration	Reference Attitude [x,y,z]	Off-Zenith Angle	Mean Visible SVs
Creamers Field	June 3, 2009	9.5 hours	[0.00, 0.00, 1.00]	0°	10.32
Urban Rooftop	June 13, 2009	6.3 hours	[0.22,-0.25, 0.94]	19°	8.15
Ester Dome	June 14, 2009	3.6 hours	[-0.68, 0.19, 0.71]	45°	8.63
Hez Ray Field	June 28, 2009	20.0 hours	[0.00, 0.00, 1.00]	0°	10.30
Hez Ray Field	July 17, 2009	18.9 hours	[-0.38, 0.00, 0.92]	22.5°	9.70

7.3 Attitude Calculations

There is a step of pre-processing that is performed on the raw measurements where a condition is placed on the SV data before it is included in the calculations: A 90 second expiry condition is implemented for each SV. This is because the Trimble GPS receiver reports the last *known* SV SNR value and LOS vector even during times when the GPS receiver temporarily loses a signal. Other times the receiver continues to report the last known SNR and LOS values after the SV has traversed below the

horizon with no actual signal being received. Therefore if either the SNR or LOS vector does not change for a particular SV during any 90 second window, the SV measurement is considered outdated and is excluded. The expiry condition is reset whenever the SNR or the LOS vector changes for each SV.

From Equation 13 in chapter 3, the receiving antenna off-boresight angle for each received SV signal is

$$\alpha_{(SNR, P_t, \theta, \phi)} = f\left(\frac{SNR}{P_t \cdot G_{t_\theta} \cdot L_{p_\theta} \cdot L_{a_\phi}}\right), \quad (24)$$

which accounts for SV to SV output power variations P_t , SV transmitting antenna gain pattern G_t , path loss L_p , and atmospheric losses L_a . To separately examine the effect on accuracy of each parameter included in Equation 24, the equation is redefined as a generalized expression

$$\alpha = f(P), \quad (25)$$

where P represents any combination of elements of Equation 24. Whatever parameters are included for each calculation set are applied to the specific SAP antenna SNR-to- α mapping function from Table 7 in chapter 4

$$\alpha = f(P) = -42.4P^3 + 68.7P^2 - 118.5P + 99.8. \quad (26)$$

Five separate calculation sets are presented using the data from the five measurement sessions outlined in Table 12. Only the raw normalized SNR values are used as parameters in the least squares equation for the first calculation set. The weighted least squares solution is used next, again using the raw normalized SNR values. SV radiation pattern and path loss parameters are included in the third calculation set. The forth calculation set includes SV output power variation, and the fifth includes atmospheric loss parameters. The attitude estimation results of each calculation set are expressed in terms of RMS error, 90th percentile error and maximum error.

7.3.1 Normalized SNR Calculation Set

The first attitude estimation calculation uses only the normalized SNR values to determine the off-boresight angle of each SV visible to the receiver so that

$$P = SNR_N. \quad (27)$$

The attitude is calculated using the least squares solution from Chapter 3 to solve for \mathbf{B} for each update interval, which occurs at a 1 Hz rate

$$\mathbf{A}^T \mathbf{A} \bar{\mathbf{B}} = \mathbf{A}^T \bar{\mathbf{S}}$$

$$\begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} L_1 \hat{x} & L_1 \hat{y} & L_1 \hat{z} \\ L_2 \hat{x} & L_2 \hat{y} & L_2 \hat{z} \\ \vdots & \vdots & \vdots \\ L_n \hat{x} & L_n \hat{y} & L_n \hat{z} \end{bmatrix} \begin{bmatrix} B \hat{x} \\ B \hat{y} \\ B \hat{z} \end{bmatrix} = \begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_n \end{bmatrix}, \quad (28)$$

with the α values of each SV coming from the normalized raw SNR measurements as the parameter in Equation 26. Table 13 shows the estimation accuracy of each measurement session. The zenith pointing sessions have the best accuracy, with progressively worse accuracy as the attitude deviate from the zenith direction. The exception is the Urban rooftop, which despite nearly pointing in the zenith direction has the worst RMS error because of considerable signal blockages.

Table 13: Attitude Estimation Results based on Normalized SNR Calculation Set

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Creamers 0°	2.25	3.97	8.61
Rooftop 19°	14.10	19.75	27.23
Ester Dome 45°	12.39	20.10	33.76
Hez Ray 0°	3.05	4.92	10.27
Hez Ray 22°	8.42	11.87	17.28
Mean of all Sessions	8.04	12.12	19.43

7.3.2 Weighting Matrix Calculation Set

The next calculation includes the weighting matrix of Equation 7 described in Chapter 3:

$$A^T W A \bar{B} = A^T W \bar{S}$$

$$\begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ 0 & \cdots & w_n \end{bmatrix} \begin{bmatrix} L_1 \hat{x} & L_1 \hat{y} & L_1 \hat{z} \\ L_2 \hat{x} & L_2 \hat{y} & L_2 \hat{z} \\ \vdots & \vdots & \vdots \\ L_n \hat{x} & L_n \hat{y} & L_n \hat{z} \end{bmatrix} \begin{bmatrix} B \hat{x} \\ B \hat{y} \\ B \hat{z} \end{bmatrix} =$$

$$\begin{bmatrix} L_1 \hat{x} & L_2 \hat{x} & \cdots & L_n \hat{x} \\ L_1 \hat{y} & L_2 \hat{y} & \cdots & L_n \hat{y} \\ L_1 \hat{z} & L_2 \hat{z} & \cdots & L_n \hat{z} \end{bmatrix} \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ 0 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_n \end{bmatrix}. \quad (29)$$

As before, the α values are derived from the normalized raw SNR measurements using Equation 26. Table 14 shows the attitude estimation results. The values in parenthesis denote the change from the previous calculation set. Most sessions show significant improvement with the exception of slight degradations of the Hez Ray session. The RMS error and 90% error of the Urban Rooftop showed the greatest improvement, but the maximum error of the Urban Rooftop session also had the worst decline.

Table 14: Attitude Estimation Results based on Weighting Matrix Calculation Set

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Creamers 0°	2.05 (-0.20)	3.50 (-.047)	7.39 (-1.22)
Rooftop 19°	5.96 (-8.14)	11.00 (-8.75)	49.88 (+22.65)
Ester Dome 45°	7.68 (-4.71)	12.70 (-7.40)	26.95 (-6.81)
Hez Ray 0°	3.46 (+0.41)	5.93 (+1.01)	10.97 (+0.70)
Hez Ray 22°	5.30 (-3.12)	9.26 (-2.61)	16.91 (-0.37)
Mean of all Sessions	4.89 (-3.15)	8.48 (-3.64)	22.42 (+2.99)

The large increase in maximum error is reasonably expected for the Urban Rooftop session with obstacles blocking SV signals. When the SV signals are obscured, the SNR significantly decreases resulting in an erroneous calculated off-boresight angle of

the SV. When there are many other good quality SV signals available in an overdetermined equation, the effect on the attitude calculation is minimal. However, in the rare scenario where there are relatively few available SV signals and the erroneous SV measurement is highly weighted (being at a high elevation) the attitude estimation has a large error.

7.3.3 SV Radiation Pattern and Path Loss Correction Calculation Set

The next calculation set includes the effects of the transmitting SV radiation pattern coefficients of Table 3 and path loss of Equation 17 as parameters

$$P = \frac{SNR_N}{Gt_\theta \cdot Lp_\theta} . \quad (30)$$

The off-boresight angle α of Equation 26 is then used in Equation 29 as the previous calculation.

Table 15 shows the estimation results, where the RMS error is modestly improved for all sessions except for a slight degradation of the Creamers session. The estimation accuracy still follows the trend of worse accuracy with attitudes farther away from zenith.

Table 15: Attitude Estimation Results based on SV Radiation Pattern and Path Loss Correction Calculation Set

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Creamers 0°	2.48 (+0.43)	4.11 (+0.61)	9.27 (+1.88)
Rooftop 19°	5.06 (-0.90)	9.36 (-1.64)	39.98 (-9.90)
Ester Dome 45°	6.04 (-1.64)	9.10 (-3.60)	28.72 (+1.77)
Hez Ray 0°	2.94 (-0.52)	4.98 (-0.95)	9.44 (-1.53)
Hez Ray 22°	4.48 (-0.82)	7.74 (-1.52)	15.97 (-0.94)
Mean of all Sessions	4.20 (-0.69)	7.06 (-1.42)	20.68 (-1.74)

7.3.4 SV Output Power Correction Calculation Set

The SV to SV output power variation is compensated in the next calculation set according to Table 1. The P_t values were determined using data from the Hez Ray 0° session. The transmitting SV radiation pattern and path loss parameters are also included in Equation 26 as in the previous calculation using

$$P = \frac{SNR_N}{G_{t_\theta} \cdot L_{p_\theta} \cdot P_t} \cdot \quad (31)$$

Table 16 shows the estimation results. The biggest improvement is with the maximum error, with little difference of the RMS and 90% errors.

Table 16: Attitude Estimation Results based on SV Output Power Correction Calculation Set

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Creamers 0°	2.40 (-0.08)	4.19 (+0.08)	8.87 (-0.40)
Rooftop 19 °	5.37 (+0.31)	10.08 (+0.72)	32.69 (-7.29)
Ester Dome 45°	4.90 (-1.14)	7.81 (-1.29)	19.56 (-9.16)
Hez Ray 0°	2.80 (-0.14)	5.09 (+0.11)	10.56 (+1.12)
Hez Ray 22°	4.40 (-0.08)	7.35 (-0.42)	13.74 (-2.19)
Mean of all Sessions	3.97 (-0.23)	6.90 (-0.16)	17.08 (-3.60)

7.3.5 Atmospheric Loss Correction Calculation Set

Atmospheric loss, occurring predominately at the lower elevation extremes, is included in the next calculation set according to Equation 18. As in the previous calculation, the transmitting SV radiation pattern, path loss and output power parameters are also included in Equation 26 using

$$P = \frac{SNR_N}{G_{t_\theta} \cdot L_{p_\theta} \cdot P_t \cdot L_{a_\phi}} \cdot \quad (32)$$

Most sessions exhibit a slight improvement in attitude estimation, but the overall effect is almost negligible as shown in Table 17. The weighting matrix, which places

less weight to signals at lower elevations, diminishes the importance of the atmospheric loss parameter.

Table 17: Attitude Estimation Results based on Atmospheric Loss Correction Calculation Set

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Creamers 0°	2.49 (+0.09)	4.35 (-0.16)	9.15 (+0.27)
Rooftop 19°	5.32 (-0.05)	9.98 (-0.10)	31.69 (-1.00)
Ester Dome 45°	4.81 (-0.09)	7.73 (-0.08)	19.75 (+0.19)
Hez Ray 0°	2.76 (-0.04)	5.03 (-0.06)	10.74 (+0.18)
Hez Ray 22°	4.34 (-0.06)	7.27 (-0.08)	13.73 (-0.01)
Mean of all Sessions	3.94 (-0.03)	6.87 (-0.03)	17.01 (-0.07)

7.4 Calculation Set Summary

Table 18 summarizes the effect the inclusion of the individual calculation factors have on the accuracy of the attitude estimation. The error values are the mean of all the measurement sessions. Starting with the raw normalized SNR, the accuracy improves with each subsequent calculation. The weighted matrix inclusion had the largest effect on the attitude estimation accuracy with almost a 40% improvement of the RMS error. The weighted matrix solution degraded the maximum error largely because of signal blockages with the Urban Rooftop session. The remaining calculation sets which incrementally included more signal parameters also improved the attitude estimation accuracy, though not to the degree as the weighted matrix.

Table 18: Mean Estimation Results for Calculation Sets

Calculation Set	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)
Raw SNR	8.04	12.12	19.43
Weighted Matrix Raw SNR	4.89 (-3.15)	8.48 (-3.64)	22.42 (+2.99)
SV Pattern and Path Loss	4.20 (-0.69)	7.06 (-1.42)	20.68 (-1.74)
SV Output Power	3.97 (-0.23)	6.90 (-0.16)	17.08 (-3.60)
Atmospheric Loss	3.94 (-0.03)	6.87 (-0.03)	17.01 (-0.07)

In [5], Lightsey assumes the signal parameters are smaller in magnitude than the noise seen in the SNR measurement, with no significant improvement to be expected from incorporating them into the measurement model. However the errors of the attitude estimation are reduced by about 20% by including the SV gain pattern, path loss and SV output power parameters in the calculation compared to using the raw SNR values, making their inclusion significant and worthwhile.

7.5 Comparison to Previous Results

The attitude estimation results of this study are compared in Table 19 to the results presented by Behre in [7]. Behre provides the only other known ground based experiment using a single antenna to compute a two-axis attitude solution. There are 5 ground experiments in [7] which all have zenith pointing antennas: The Ground TANS experiment used a Trimble TANS Vector receiver and a standard Trimble patch antenna. The remaining 4 experiments all used a NovAtel 12 channel receiver. NovAtel #1 used a NovAtel Patch antenna, NovAtel #2 used an Ashtech choke ring antenna, NovAtel #3 a standard Trimble patch antenna, and NovAtel #4 used the same NovAtel Patch as NovAtel #1.

Table 19: Zenith Pointing Attitude Estimation Results for Ground Based Measurements

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)	Mean Visible SVs
TANS	5.0	7.5	22.7	5.8
NovAtel #1	6.1	9.4	25.1	8.2
NovAtel #2	4.2	6.4	21.5	8.2
NovAtel #3	5.7	8.6	33.8	8.2
NovAtel #4	6.0	9.1	18.5	7.9
Testbed Creamers	2.5	4.5	9.2	10.3
Testbed Hez Ray	2.8	5.3	10.7	9.7

The attitude estimation method in [7] uses a weighted least squares calculation similar to this study, except the elements of the weighting matrix are assigned to the offboresight angle variance values obtained from the data samples of the SNR-to- α mapping function. The attitude estimations also include SV gain pattern and space

loss parameters in the calculations, though the SV gain pattern models differed from the ones used in this study.

The two zenith pointing measurement sessions in this study have 33% to 59% lower estimation RMS errors compared to the experiments in [7]. It is apparent in Table 19 that there were significantly more SVs in the GPS constellation in 2009 when the measurements were taken for this study compared to 1997 for [7], which undoubtedly contributes to the more accurate attitude estimation of this study. The improvement may also be partly attributed to a better suited antenna, better SV gain pattern modeling, or the type of weighting matrix used in the least squares equation.

To gauge the type of performance the testbed hardware might have in orbit, the results from the GANE orbital experiment in [3] are compared to the ground TANS experiment in [7] where the same GPS hardware was used. Two GANE measurement sessions were performed on board the Space Shuttle *Endeavour* in 1996. The first had the antenna oriented in the zenith direction, and the second at 45° off-zenith. The trends in the GANE experiments are similar to this study, where the zenith measurements are the most accurate and become degraded for other angles as shown in Table 20.

Table 20: Attitude Estimation Results for TANS Hardware Measurements by Behre

Session	RMS Error (Deg)	90% Error (Deg)	Maximum Error (Deg)	Mean Visible SVs
Ground TANS 0°	5.0	7.5	22.7	5.8
GANE TANS 0°	3.2	4.9	9.9	5.4
GANE TANS 45°	5.2	8.3	24.0	3.8

Using the same TANS hardware for ground and space based experiments, the estimation results for the zenith measurements are more accurate in orbit compared to the ground, feasibly due to less occurrence of signal multipath. The same may also hold true for the testbed hardware using the SAP antenna.

Chapter 8: Conclusion and Future Work

A specialized SAP antenna was designed and built to perform ground based GPS SNR measurements for static attitude estimation. The SAP antenna electrical characteristics were measured, with performance as expected within the design criteria. The smaller size and lighter weight of the SAP antenna compared to the survey grade choke ring antenna with similar performance makes the SAP antenna well suited for small satellite GPS attitude determination.

Three PCB revisions were designed and built during the development of the embedded testbed hardware. A new PIPC inter-processor DMA communication protocol was developed allowing for energy efficient high performance parallel processing. Performance measurements of the PIPC protocol showed it is up to 32 times faster and consumes three times less current than the alternate SPI-based IPC protocol. The newly developed protocol is applicable to designs other than this thesis where multi-processing is used in embedded hardware.

The ADS accuracy results for the ground zenith measurements were approximately twice as accurate compared to previous studies where a single antenna was used to estimate a two-axis attitude. Off-zenith angle measurements were also improved compared to previous results. The estimation improvement was due in part to more available SV signals than in the previous studies, but may also be partly attributed to a better suited antenna, better SV gain pattern modeling, or the type of weighting matrix used in the least squares equation.

There is ample room for future refinement of the SNR ADS testbed system. The SAP antenna radiation pattern should be precisely characterized in an anechoic chamber instead of collecting SNR samples from a known attitude. A precisely known antenna radiation pattern used to generate SNR-to- α mapping function would allow for specific modeling of the individual SV antenna patterns and output power levels, potentially increasing the ADS estimation accuracy further.

Several different types GPS modules could be simultaneously tested to compare accuracy using the hardware already developed. Similarly, other configurations of the SAP antenna could be simultaneously compared based on specific CubeSat size constraints. A pinwheel type antenna should also be evaluated based on its theoretical performance characteristics. Dynamic measurements should also be evaluated before implementation of the GPS ADS on a CubeSat. This could be accomplished by mounting the system on an Unmanned Aerial Vehicle with a precise reference AD system. Finally, a suitable GPS module needs to be verified for orbit operation, either through hardware-in-the-loop simulation or by testing beyond COCOM speed and altitude limits.

A GPS based attitude determination system is recommended for consideration on a CubeSat design based on preliminary ground measurements. For a case where an orbit determination GPS system is already an implemented subsystem on a CubeSat design, no extra hardware is required for attitude determination— only some additional processing resources are required. The only specialized hardware design consideration from the typical GPS implementation is the use of an optimal antenna with a monotonic radiation pattern.

References

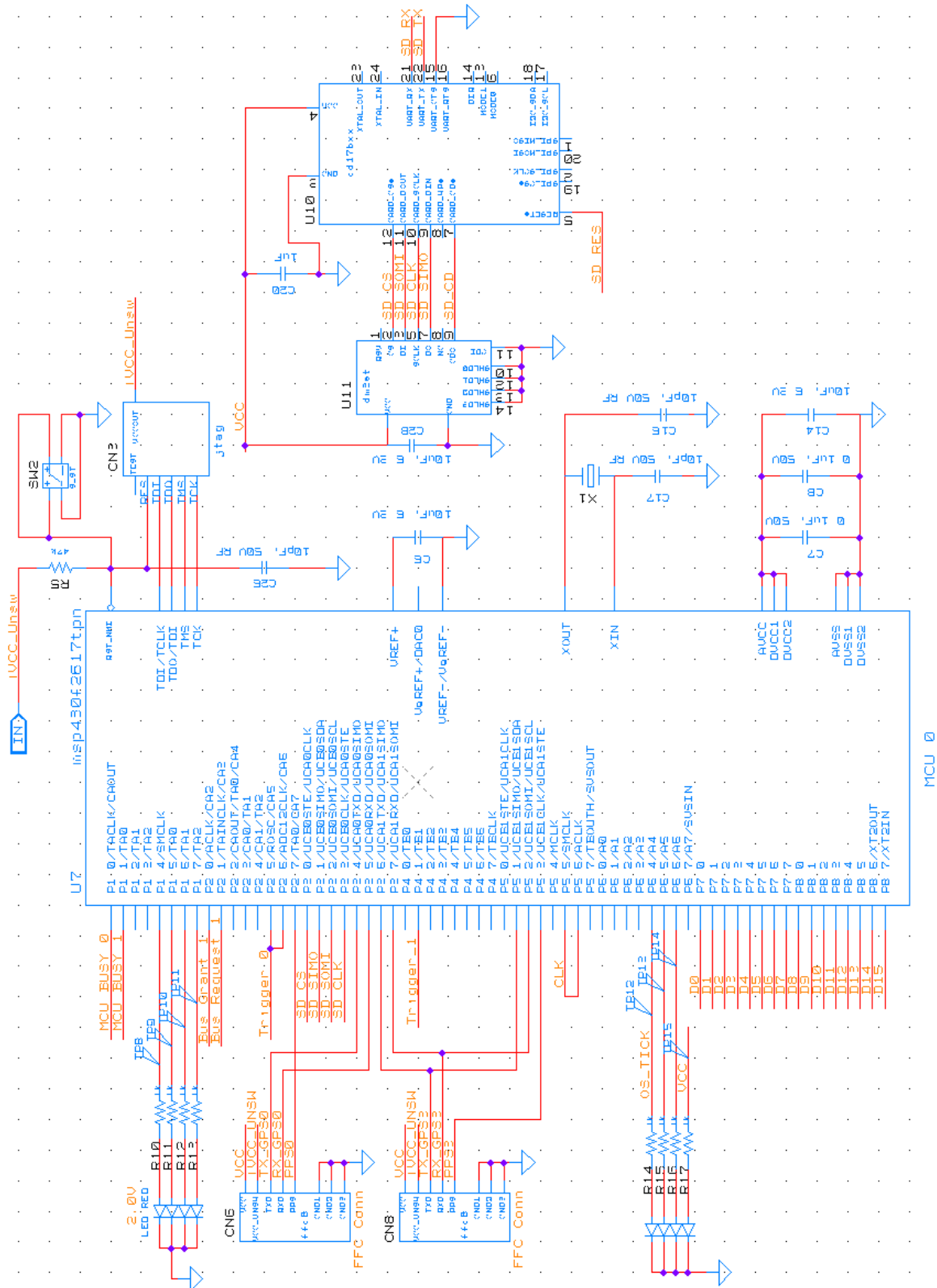
- [1] O. Kurz, “Design and Implementation of an Attitude Determination System for the Cubesat UWE-2”, Masters Thesis, University of Wuerzburg, Germany, December 2007.
- [2] A. Aydinlioglo and M. Hammer, “COMPASS- 1 Pico Satellite: Magnetic Coils for Attitude Control”, Proceedings of 2nd International Conference on Recent Advances in Space Technologies, Istanbul, Turkey, June 9-11, 2005.
- [3] P. Axelrad and C. P. Behre, “Satellite Attitude Determination Based on GPS Signal-to-Noise Ratio”, Proceedings of the IEEE, Vol. 87, No. 1, January 1999.
- [4] C. Wang. et al, “An Improved Single Antenna Attitude System Based on GPS Signal Strength”, AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, August 15–18, 2005.
- [5] E. Glenn Lightsey and Jared D. Madsen. “Three-Axis Attitude Determination Using Global Positioning System Signal Strength Measurements”, AIAA Journal of Guidance, Control and Dynamics, March-April 2003.
- [6] J. Madsen, “Robust Spacecraft Attitude Determination Using Global Positioning System Receivers”, Dissertation, University of Texas, Austin, TX, May 2003.
- [7] C. Behre, “GPS Based Algorithms For Low Cost Satellite Missions, Dissertation, University of Colorado, Boulder, CO, December 1997.
- [8] UBlox AG, “The GPS Dictionary”, GPS-X-00001, March 8, 2001.
- [9] E. Rogstad, “Testing and Development of the UT Orion GPS Receiver”, Masters Thesis, The University of Texas at Austin, August 2008.
- [10] J. Helmericks, “Differential GPS Tracking of Sounding Rocket Payloads”, Masters Thesis, University of Alaska, Fairbanks, AK, August 2001.

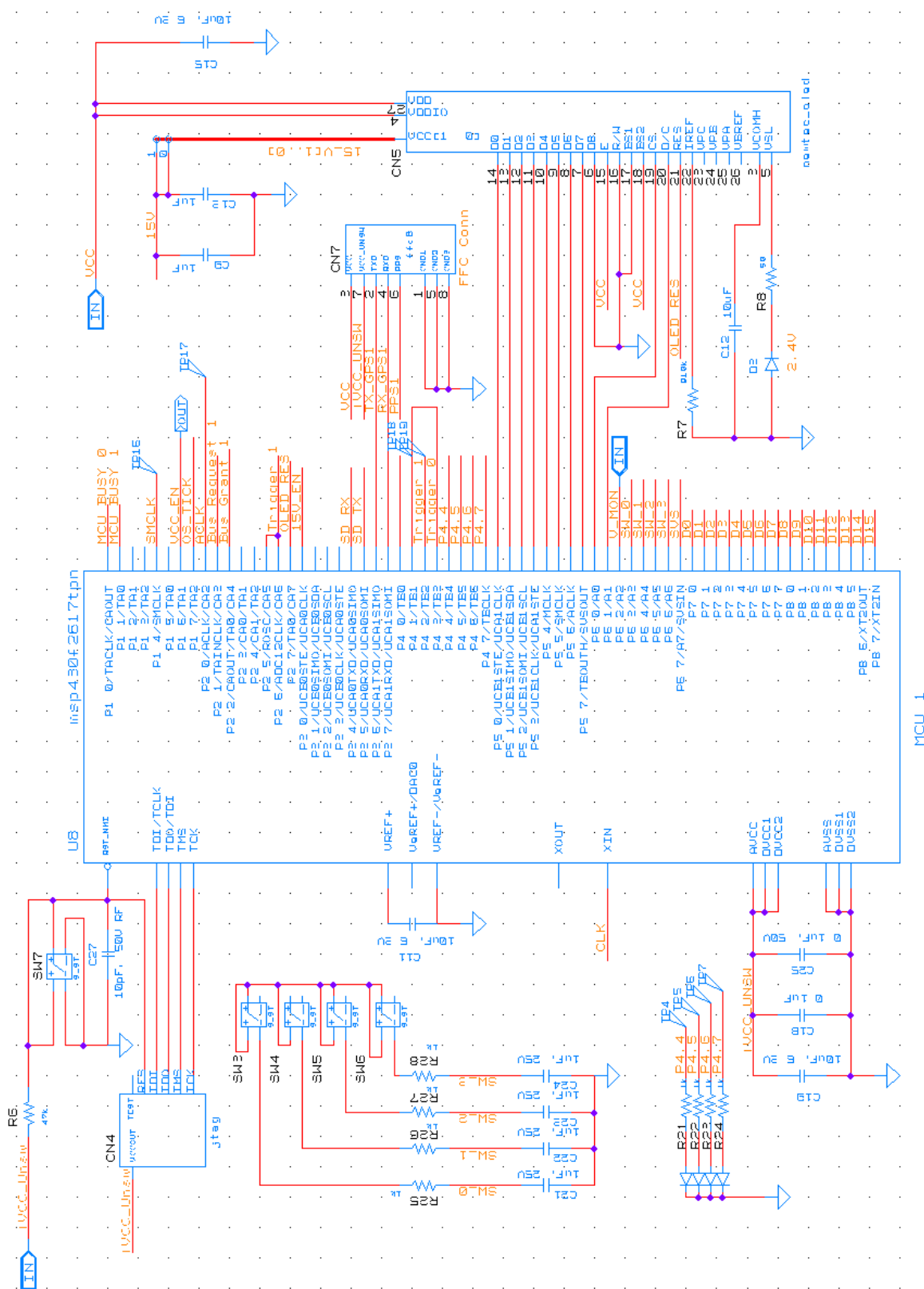
- [11] S. Purivigraipong et al., "New Approach to Achieving Stand Alone GPS Attitude Determination using Dual Short Baselines for Small-Satellite", 13th Annual AIAA/USU Conference on Small Satellites, SSC99-XI-2, August 26, 1999.
- [12] G. Strang and K. Borre, "Linear Algebra, Geodesy, and GPS", Wellesley-Cambridge Press, 1997.
- [13] E. Kaplan and C. Hegarty, "Understanding GPS Principles and Applications—2nd Ed", Artech House Inc, 2006.
- [14] F. Czopek, "Description and Performance of the GPS Block I and II L-Band Antenna and Link Budget", Proceedings of the ION GPS 93, Salt Lake City, UT, September 22 - 24, 1993.
- [15] A. Wu, "Predictions and Field Measurements of the GPS Block IIR L1 and L2 Ground Powers", ION NTM 2002, San Diego, CA, January 28-30, 2002.
- [16] W. Marquis, "On-Orbit Performance of the Improved GPS Block IIR Antenna Panel", Proceedings of the 18th International Technical Meeting of the Satellite Division of the ION GNSS, Long Beach, CA, September 13-16, 2005.
- [17] C. Wang. et al, "A GPS Signal Transmission Model for Improved Single Antenna Attitude Determination", Queensland University of Technology, Australia, 2005.
- [18] M. Moreau, "Results from the GPS Flight Experiment on the High Earth Orbit AMSAT OSCAR-40 Spacecraft", Proceedings of the 15th International Technical Meeting of the Satellite Division of the ION GPS 2002, Portland, OR, September 24 - 27, 2002.
- [19] J. Kouba, "A Simplified Yaw-Attitude Model For Eclipsing GPS Satellites", GPS Solutions, Vol. 13, January 2009, pp 1-12.
- [20] J. Um and G. Lightsey, "GPS Attitude Determination for the SOAR Experiment", Navigation, Vol. 48, No. 3, Fall 2001, pp. 181-194.
- [21] D. Jackson, "Chapter 7: Microstrip Antennas", Antenna Engineering Handbook 4th Ed., ed. J. Volakis (New York: McGraw-Hill, 2007.), pp 27.

- [22] L. Basilio et al., "A Comparative Study of a New GPS Reduced-Surface Wave Antenna", IEEE Antennas and Wireless Propagation Letters, Vol. 4, 2005
- [23] L. Boccia et al, "Shorted annular patches as flexible antennas for space applications", 17th International Conference on Applied Electromagnetics and Communications, ICECom, October 1-3, 2003.
- [24] A. Zaghloul, "Dual and Tri Band Hybrid Phased Array Systems", IDGA 2005 Military Antenna Systems, Arlington, VA, September 20–21, 2005.
- [25] G. Moernaut and G. Vandenbosch, "The Shorted Annular Patch As a Multipath Suppressing Antenna", The Second European Conference on Antennas and Propagation, Edinburgh, UK, Nov 11-16 2007.
- [26] L. Boccia et al., " Performance Evaluation of Shorted Annular Patch Antennas for High-Precision GPS Systems", The Institution of Engineering and Technology, Microwaves, Antennas & Propagation, Vol. 1, Issue 2, April 2007 pp 465 - 471.
- [27] D. Peters, D. Raskovic, and D. Thorsen , "An Energy Efficient Parallel Embedded System for Small Satellite Applications," ISAST Transactions on Computers and Intelligent Systems, vol. 1(2), 2009, pp. 8-16.
- [28] Septentrio Satellite Navigation PolaRx2/2e User Manual, V3.2.2, November 27, 2008.



A.2 Testbed Version 2.1 MCU0 Circuits





Appendix B: Attitude Determination MATLAB Code

B.1 MATLAB Constant Declarations

```
% constants.m
% Loads constants to workspace for Attitude Estimation Calculations

max_AMU=16.3;           % max recorded SNR value (AMUs)
max_dBHz = 27 + 20*log10(max_AMU); % max recorded SNR value (dB-Hz)
orbit_altitude=0; %user orbit (km)

max_snr=zeros(32,1,'single');

max_snr(:) = max_AMU;

max_snr(1) =max_AMU*1.00;
max_snr(2) =max_AMU*0.97;
max_snr(3) =max_AMU*0.95;
max_snr(4) =max_AMU*0.90;
max_snr(5) =max_AMU*1.00; %currently the only non-active PRN
max_snr(6) =max_AMU*0.91;
max_snr(7) =max_AMU*0.96;
max_snr(8) =max_AMU*0.88;
max_snr(9) =max_AMU*0.97;
max_snr(10)=max_AMU*0.89;
max_snr(11)=max_AMU*1.08;
max_snr(12)=max_AMU*0.98;
max_snr(13)=max_AMU*0.97;
max_snr(14)=max_AMU*0.87;
max_snr(15)=max_AMU*0.99;
max_snr(16)=max_AMU*0.97;
max_snr(17)=max_AMU*0.92;
max_snr(18)=max_AMU*0.93;
max_snr(19)=max_AMU*0.95;
max_snr(20)=max_AMU*0.93;
max_snr(21)=max_AMU*0.95;
max_snr(22)=max_AMU*1.00;
max_snr(23)=max_AMU*1.07;
max_snr(24)=max_AMU*1.00;
max_snr(25)=max_AMU*0.97;
max_snr(26)=max_AMU*0.96;
max_snr(27)=max_AMU*0.85;
max_snr(28)=max_AMU*0.88;
max_snr(29)=max_AMU*0.96;
max_snr(30)=max_AMU*1.03;
max_snr(31)=max_AMU*1.01;
max_snr(32)=max_AMU*0.93;

SV_Tx_dB(1) = 0.00;
SV_Tx_dB(2) = -0.25;
SV_Tx_dB(3) = -0.45;
SV_Tx_dB(4) = -0.90;
SV_Tx_dB(5) = 0.00; %currently the only non-active PRN
SV_Tx_dB(6) = -0.80;
SV_Tx_dB(7) = -0.35;
SV_Tx_dB(8) = -1.10;
SV_Tx_dB(9) = -0.25;
SV_Tx_dB(10)= -1.10;
SV_Tx_dB(11)= 0.65;
SV_Tx_dB(12)= -0.20;
SV_Tx_dB(13)= -0.25;
SV_Tx_dB(14)= -1.20;
SV_Tx_dB(15)= -0.10;
SV_Tx_dB(16)= -0.25;
SV_Tx_dB(17)= -0.70;
SV_Tx_dB(18)= -0.65;
SV_Tx_dB(19)= -0.45;
SV_Tx_dB(20)= -0.65;
SV_Tx_dB(21)= 0.45;
SV_Tx_dB(22)= 0.00;
SV_Tx_dB(23)= 0.60;
SV_Tx_dB(24)= 0.00;
SV_Tx_dB(25)= -0.25;
```



```

SV_Tx_dB(26)= -0.35;
SV_Tx_dB(27)= -1.40;
SV_Tx_dB(28)= -1.10;
SV_Tx_dB(29)= -0.35;
SV_Tx_dB(30)=  0.25;
SV_Tx_dB(31)=  0.10;
SV_Tx_dB(32)= -0.65;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%sv_block key:
%1=block II / IIA
%2=block IIR Legacy Antenna
%3=block IIR / II-M Improved Antenna
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

SV_block=zeros(32,1,'uint8');

SV_block(32)=1;
SV_block(24)=1;
SV_block(25)=1;
SV_block(26)=1;
SV_block(27)=1;
SV_block(09)=1;
SV_block(05)=1;
SV_block(04)=1;
SV_block(06)=1;
SV_block(03)=1;
SV_block(10)=1;
SV_block(30)=1;
SV_block(08)=1;

SV_block(13)=2;
SV_block(11)=2;
SV_block(20)=2;
SV_block(28)=2;
SV_block(14)=2;
SV_block(18)=2;
SV_block(16)=2;
SV_block(21)=2;

SV_block(22)=3;
SV_block(19)=3;
SV_block(23)=3;
SV_block(02)=3;
SV_block(29)=3;
SV_block(15)=3;
SV_block(31)=3;
SV_block(07)=3;
SV_block(12)=3;
SV_block(17)=3;
SV_block(01)=3;

E=6370; %Earth radius
S=26560; %SV radius
O=6; %equivalent oxygen height
a=O/E;

phi2theta=asin((cosd(0:0.1:90)*(E+orbit_altitude))/S)*180/pi;
theta=[max(phi2theta):-max(phi2theta)/900:0];

d=S*cosd(phi2theta)-(((E+orbit_altitude)^2)-((S^2)*(sind(phi2theta)).^2)).^0.5;

space_loss=1./(4*pi*d.^2);
space_loss_dB=10*log10(space_loss/max(space_loss)); %convert to normalized dB
space_loss = 10.^(space_loss_dB/20); %convert normalized dB to AMU
atm_loss_dB=-0.07*(1+2*a)./(sind(0:0.1:90)+((sind(0:0.1:90)).^2+2*a+a^2).^0.5);
atm_loss = 10.^(atm_loss_dB/20); %convert normalized dB to AMU

% Pattern Coefficients:
p1 = -3.1653e-019;
p2 = 5.748e-006;
p3 = -0.0040676;
p4 = 0.061662;
p5 = -0.019623;
p6 = -0.0066788;

```

```

SV_pattern_1_dB = (p1*phi2theta.^5 + p2*phi2theta.^4 + p3*phi2theta.^3 + p4*phi2theta.^2 +
p5*phi2theta + p6);
SV_pattern_1 = 10.^(SV_pattern_1_dB/20);
SV_power_1_dB = -(SV_pattern_1_dB + space_loss_dB );
SV_power_1 = 10.^(SV_power_1_dB/20);

% Pattern Coefficients:
p1 = 2.5085e-005;
p2 = -0.0011136;
p3 = 0.010073;
p4 = 0.0044916;
p5 = 0.035453;
p6 = -0.051893;

SV_pattern_2_dB = (p1*phi2theta.^5 + p2*phi2theta.^4 + p3*phi2theta.^3 + p4*phi2theta.^2 +
p5*phi2theta + p6);
SV_pattern_2 = 10.^(SV_pattern_2_dB/20);
SV_power_2_dB=-(SV_pattern_2_dB + space_loss_dB );
SV_power_2=10.^(SV_power_2_dB/20);

% Pattern Coefficients:
p1 = 5.8997e-005;
p2 = -0.0028536;
p3 = 0.040936;
p4 = -0.18848;
p5 = 0.14967;
p6 = 1.1324;

SV_pattern_3_dB = (p1*phi2theta.^5 + p2*phi2theta.^4 + p3*phi2theta.^3 + p4*phi2theta.^2 +
p5*phi2theta + p6);
SV_pattern_3 = 10.^(SV_pattern_3_dB/20);
SV_power_3_dB=-(SV_pattern_3_dB + space_loss_dB );
SV_power_3=10.^(SV_power_3_dB/20);

clear p1 p2 p3 p4 p5 p6 O E S a orbit_altitude

```

B.2 MATLAB Script for Loading Testbed GPS Log File

```

% load_testbed.m
% Loads Testbed GPS SNR data samples to workspace

[FileName,PathName] = uigetfile('*.gps','Open GPS Attitude Determination Testbed Log File');
if FileName == 0
    return;
end

fid = fopen(fullfile(PathName,FileName));
if fid == -1
    return;
end

disp(['Opening ' FileName] );
SATPacket=struct('SNR',single(0),'elevation',single(0),'azimuth',single(0));
SDPacket=struct('GPSTime',uint32(0),'SATPacket',repmat(SATPacket,32,1));

fseek(fid,0,'eof');
f_length=ftell(fid);
if f_length == -1
    return;
end

testbed=repmat(SDPacket,floor(f_length/200),1);
fseek(fid,0,'bof');

n=1; % index
m=1; % index
packetcount=0;
badpacket=0;
GPSTIMEPREV=NaN;

h = waitbar(0,'Reading File...');

while( ftell(fid)<f_length )
    if mod(ftell(fid),10000) ==0

```

```

        waitbar(ftell(fid)/f_length)
    end
    STX=0;
    while (STX~=hex2dec('55') && ftell(fid)<f_length)
        STX=fread(fid,1,'uint8');
        if STX==hex2dec('55')
            STX=fread(fid,1,'uint8');
        else
            badpacket=1;
        end
    end
    GPSTIME=fread(fid,1,'uint32');
    if (isnan(GPSTIMEPREV))
        GPSTIMEPREV=GPSTIME;
    end
    buffer=fread(fid,96,'uint16');
    ETX=fread(fid,1,'uint16');
    if isempty(ETX) || ETX~=hex2dec('AAAA') || badpacket==1

    else
        GPSTIMEPREV=GPSTIME;
        testbed(n).GPSTime=GPSTIME;
        m=1;
        for SV=1:32
            testbed(n).SATPacket(SV).SNR=buffer(m)/10;
            testbed(n).SATPacket(SV).elevation=(0.001*buffer(m+1)); % #'s were in rad*1000 on
testbed
            testbed(n).SATPacket(SV).azimuth=(0.001*buffer(m+2));
            m=m+3;
            if testbed(n).SATPacket(SV).SNR>30 || testbed(n).SATPacket(SV).elevation>1.5708 %look
for invalid data
                testbed(n).SATPacket(SV).SNR=0;
                testbed(n).SATPacket(SV).elevation=0;
            end
        end
        n=n+1;
        packetcount=packetcount+1;
        badpacket=0;
    end
end
fclose(fid);
close(h)
if packetcount >0
    packetcount %display packetcount in matlab workspace
else
    disp('Error: Unrecognized file or no data samples')
end
testbed_packets=packetcount;
clear ans ETX FileName PathName GPSTIME GPSTIMEPREV SATPacket SDPacket STX SV ;
clear active_SV buffer fid goodpacket m n packetcount f_length h ;

```

B.3 MATLAB Script for Loading PolaRX Reference GPS Log File

```

% load_polaRX.m

% Loads Reference Truth Attitude samples into workspace, and calculates
% mean attitude from all samples taken.

[FileName,PathName] = uigetfile('*.SBF','Open Septentrio PolaRx2eH Log File for Reference Attitude.
');
if FileName == 0
    return;
end
fid = fopen(fullfile(PathName,FileName));
if fid == -1
    return;
end
SBFPacket=struct('GPSTime',uint32(0),'elevation',single(0),'azimuth',single(0));
packetcount=0;
goodpacket=1;

while(goodpacket==1)
    sync=fread(fid,1,'uint16','ieee-le');

```

```

if isempty(sync) || sync ~= 16420
    goodpacket=0;
    break;
end
crc=fread(fid,1,'uint16','ieee-le');
blockID=fread(fid,1,'uint16','ieee-le');
if isempty(blockID) || blockID ~=5938
    goodpacket=0;
    break;
end
blocksize=fread(fid,1,'uint16','ieee-le');
if isempty(blocksize) || blocksize <8 || blocksize>4096 || mod(blocksize,4)~=0
    goodpacket=0;
    break;
end
dummy=fread(fid,blocksize-8,'uint8');
if goodpacket==1;
    packetcount=packetcount+1;
else
    goodpacket=0;
    break;
end
end

fclose(fid);
if packetcount >0
    packetcount %display packetcount in matlab workspace
else
    error('Error: Unrecognized file or no data ')
    return;
end
SBF=repmat(SBFPacket,packetcount,1);
fid = fopen(fullfile(PathName,FileName));
n=1; % index
for n=1:packetcount
    fread(fid,8,'uint8'); %read and discard
    SBF(n).GPSTime=fread(fid,1,'uint32','ieee-le');
    fread(fid,8,'uint8'); %read and discard
    SBF(n).azimuth=fread(fid,1,'single','ieee-le');
    if SBF(n).azimuth > 360 SBF(n).azimuth=0; end %out of range values
    SBF(n).elevation=90+fread(fid,1,'single','ieee-le');
    if SBF(n).elevation > 90 SBF(n).elevation=0; end %out of range values
    fread(fid,16,'uint8'); %read and discard
end
fclose(fid);

TRUE_ATT=single(zeros(packetcount,3));
[TRUE_ATT(:,1),TRUE_ATT(:,2),TRUE_ATT(:,3)]=sph2cart([SBF(:).azimuth]*pi/180,[SBF(:).elevation]*pi/180,1);

global TRUE_MEAN;

TRUE_MEAN(1)=mean(TRUE_ATT(:,1));
TRUE_MEAN(2)=mean(TRUE_ATT(:,2));
TRUE_MEAN(3)=mean(TRUE_ATT(:,3));

TRUE_std_az=std([SBF(:).azimuth]);
TRUE_std_el=std([SBF(:).elevation]);
polarx_packets=packetcount;

clear n FileName PathName SBFPacket blockID blocksize crc dummy fid goodpacket packetcount sync ans
packetcount SBF ;

```

B.4 MATLAB Script for Calculating Attitude and Presenting Statistics

```

% static_attitude.m
% Calculates Antenna Attitude from SV LOS vectors and SNR values
% Generates statistics and plots

constants; % Loads SV Groups and SV Radiation Pattern Coefficients

SNR_Mask_AMU = 1.2 / max_AMU ; %AMU minimum before measurement is used in solution
SNR_Mask = SNR_Mask_AMU; %define which units to use

```

```

%Pre-allocate variable memory
xax=single(1:testbed_packets);
SNR=zeros(32,testbed_packets,'single');
elevation=zeros(32,testbed_packets,'single');
azimuth=zeros(32,testbed_packets,'single');
SVused=zeros(1,testbed_packets,'uint8');
expired=zeros(32,1,'uint32');
B=zeros(testbed_packets,3,'single');
error=zeros(testbed_packets,1,'single');
alpha=single(nan(32,testbed_packets));
alpha_true=single(nan(32,testbed_packets));

% x-axis for plotting
% SNR values from data samples
% SV LOS vector elevation
% SV LOS vector azimuth
% # of SVs used in Calculation
% Flag to determine stale SNR measurement of SV
% Attitude Solution in Cartesian Coordinates
% Attitude Estimation error
% Calculated SV off-boresight angle from SNR
% TRUE SV off-boresight angle

true_att=TRUE_MEAN;

h = waitbar(0,'Processing Data...');

%antenna pattern mapping function polynomial coefficients
% Valid for SNR in AMU units, [normalized].
p4= 99.7649;
p3=-118.4877;
p2= 68.6685;
p1= -42.3644;

% ignore first 5 samples from GPS startup
for n=5:testbed_packets
    if mod(n,1000) ==0
        waitbar(n/testbed_packets)
    end
    L=single(zeros(32,3)); %satellite LOS vectors
    S=single(zeros(32,1)); %Normalized SNR values
    for SV=1:32
        elevation(SV,n)=testbed(n).SATPacket(SV).elevation;
        azimuth(SV,n)=testbed(n).SATPacket(SV).azimuth;

        if exist('Include_SV_Power_Compensation','var')
            SNR(SV,n)=((testbed(n).SATPacket(SV).SNR)/max_snr(SV));
        else
            SNR(SV,n)=((testbed(n).SATPacket(SV).SNR)/max_AMU);
        end

        if exist('Include_SV_Radiation_Pattern_Compensation','var')
            if SV_block(SV)==1 %block II/IIA
                SNR(SV,n)=SNR(SV,n)/SV_pattern_1(floor(elevation(SV,n)*1800/pi)+1);
            elseif SV_block(SV)==2 %block IIR Legacy
                SNR(SV,n)=SNR(SV,n)/SV_pattern_2(floor(elevation(SV,n)*1800/pi)+1);
            elseif SV_block(SV)==3 %block IIR Improved Antenna / IIR-M
                SNR(SV,n)=SNR(SV,n)/SV_pattern_3(floor(elevation(SV,n)*1800/pi)+1);
            end
        end

        if exist('Include_Space_Loss_Compensation','var')
            SNR(SV,n)=SNR(SV,n)/space_loss(floor(elevation(SV,n)*1800/pi)+1);
        end

        if exist('Include_Atm_Loss_Compensation','var')
            SNR(SV,n)=SNR(SV,n)/atm_loss(floor(elevation(SV,n)*1800/pi)+1);
        end
        % Increment expired counter for non-changing SV LOS vector or SNR
        if ((elevation(SV,n) - elevation(SV,n-1)) == 0) && ((azimuth(SV,n) - azimuth(SV,n-1)) == 0)
            expired(SV)=expired(SV)+1;
        else
            expired(SV)=0;
        end
        % Only use SV data in calculation if data is not stale, above horizon and above SNR
        threshold
        if expired(SV)<90 && SNR(SV,n)> SNR_Mask && elevation(SV,n) >= 0.02 % 0.02 radians
            [L(SV,1),L(SV,2),L(SV,3)]=sph2cart(azimuth(SV,n),elevation(SV,n),1);
            alpha(SV,n) = (p1*SNR(SV,n)^3 + p2*SNR(SV,n)^2 + p3*SNR(SV,n)^1 + p4);
            S(SV) = cosd(alpha(SV,n));
            if S(SV) < 0 || alpha(SV,n) > 90
                S(SV) = nan;
                L(SV,:)= [0 0 0];
                SNR(SV,n)=nan;
            end
        end
    end
end

```

```

        else
            SVused(n)=SVused(n)+1;
        end
    else
        SNR(SV,n)=nan;
        alpha(SV,n)=nan;
    end
end

alpha_true(SV,n)=(acos((dot(L(SV,1:3),true_att'))/(norm(L(SV,1:3))*norm(true_att')))*180/pi);

end

if exist('Include_Weight_Matrix','var')
    W=single(diag(ones(1,32)));
    for SV=1:32
        if SNR(SV,n)>SNR_Mask
            W(SV,SV)=cosd(alpha(SV,n))^2 + sin(elevation(SV,n))^2;
        else
            W(SV,SV)=0;
        end
    end
    B(n,:)=(L'*W*L)\(L'*W*S); % Weighted Matrix attitude calculation
else
    B(n,:)=(L'*L)\(L'*S); % Least Squares attitude calculation
end
if max(isnan(B(n,:)))>0 || SVused(n)<3
    B(n,:)=B(n-1,:);
end
end

for n=1:testbed_packets
    error(n)=(subspace(B(n,:),true_att(:))*180/pi;
end
close(h)

figure('Name','Block II/IIA SNR Levels' );
plot_n=1;
hold on;
for SV=1:32
    if max(SNR(SV,:))>0.25 && SV_block(SV)==1
        subplot(4,3,plot_n);
        plot_n=plot_n+1;
        hold on;
        plot(xax,alpha(SV,:),xax,alpha_true(SV,:),'-r','LineWidth',2);
        set(gca,'YTick',[0 30 60 90],'XTick',[0
testbed_packets],'xlim',[0,testbed_packets],'XTickLabel',[0 testbed_packets],'ylim',[0,90]);
        title(['PRN ' int2str(SV)],'fontweight','b');
        xlabel('Time (s)');
    end
end

figure('Name','Block IIR Legacy SNR Levels' );
plot_n=1;
hold on;
for SV=1:32
    if max(SNR(SV,:))>0.25 && SV_block(SV)==2
        subplot(4,3,plot_n);
        plot_n=plot_n+1;
        hold on;
        plot(xax,alpha(SV,:),xax,alpha_true(SV,:),'-r','LineWidth',2);
        set(gca,'YTick',[0 30 60 90],'XTick',[0
testbed_packets],'xlim',[0,testbed_packets],'XTickLabel',[0 testbed_packets],'ylim',[0,90]);
        title(['PRN ' int2str(SV)],'fontweight','b');
        xlabel('Time (s)');
    end
end

figure('Name','Block IIR(New) / IIR-M SNR Levels' );
plot_n=1;
hold on;
for SV=1:32
    if max(SNR(SV,:))>0.25 && SV_block(SV)==3
        subplot(4,3,plot_n);
        plot_n=plot_n+1;
        hold on;

```

```

        plot(xax,alpha(SV,:),xax,alpha_true(SV,:),'-r','LineWidth',2);
        set(gca,'YTick',[0 30 60 90],'XTick',[0
testbed_packets],'xlim',[0,testbed_packets],'XTickLabel',[0 testbed_packets],'ylim',[0,90]);
        title(['PRN ' int2str(SV)],'fontweight','b');
        xlabel('Time (s)');
    end
end

error(1:5)=error(6); %ignore 1st 5 samples from GPS startup
rms=mean(sqrt(error.^2));
[histo,bin]=hist(error,max(error)*100);
n=1;
m=1;
while m<testbed_packets*0.9
    m=sum(histo(1:n));
    n=n+1;
end
ninety=bin(n);

max_error=max(error);
meanSV=mean(SVused);
AVG_att = mean(B);
AVG_error = (acos((dot(AVG_att,true_att'))/(norm(AVG_att)*norm(true_att')))*180/pi);
disp('    RMS    Ninety    Max    Mean SVs    AVG ')
disp([rms    ninety max_error meanSV AVG_error])
figure
hold on
hist(error,max(error)*2);
grid on;
xlabel('Error (Deg)');
ylabel('Sample Quantity');

figure('Position',[1 1 1280 200])
hold on;
plot(error);
xlabel('Sample (s)');
ylabel('Error (Deg)');

figure
hold on
AX=plot(B(10:end,1),B(10:end,2));
plot(TRUE_MEAN(1),TRUE_MEAN(2),'or','MarkerFaceColor','r')
xlabel('X-Axis');
ylabel('Y-Axis');

clear AX H1 H2 BX histo bin n m

```

B.5 MATLAB Script for Generating User Interface Program

```
% Calculate_Attitude.m

% Launches GUI to load reference attitude, load GPS SNR sample data,
% calculate attitude, and generate plots

function varargout = Calculate_Attitude(varargin)
% CALCULATE_ATTITUDE M-file for Calculate_Attitude.fig
%   CALCULATE_ATTITUDE, by itself, creates a new CALCULATE_ATTITUDE or raises the existing
%   singleton*.
%
%   H = CALCULATE_ATTITUDE returns the handle to a new CALCULATE_ATTITUDE or the handle to
%   the existing singleton*.
%
%   CALCULATE_ATTITUDE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in CALCULATE_ATTITUDE.M with the given input arguments.
%
%   CALCULATE_ATTITUDE('Property','Value',...) creates a new CALCULATE_ATTITUDE or raises
%   the existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Calculate_Attitude_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Calculate_Attitude_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Calculate_Attitude

% Last Modified by GUIDE v2.5 15-Jun-2013 14:36:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Calculate_Attitude_OpeningFcn, ...
                  'gui_OutputFcn',  @Calculate_Attitude_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Calculate_Attitude is made visible.
function Calculate_Attitude_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Calculate_Attitude (see VARARGIN)

% Choose default command line output for Calculate_Attitude
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

initialize_gui(hObject, handles, false);

% UIWAIT makes Calculate_Attitude wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Calculate_Attitude_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
```



```

% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% -----
function initialize_gui(fig_handle, handles, isreset)
% If the metricdata field is present and the reset flag is false, it means
% we are just re-initializing a GUI by calling it from the cmd line
% while it is up. So, bail out as we dont want to reset the data.
global TRUE_MEAN;
TRUE_MEAN(1) = str2double(get(handles.editX, 'String'));
TRUE_MEAN(2) = str2double(get(handles.editY, 'String'));
TRUE_MEAN(3) = str2double(get(handles.editZ, 'String'));
assignin('base', 'TRUE_MEAN', TRUE_MEAN)
if isfield(handles, 'metricdata') && ~isreset
    return;
end

% Update handles structure
guidata(handles.figure1, handles);

% --- Executes on button press in pushbuttonPolarX.
function pushbuttonPolarX_Callback(hObject, eventdata, handles)
% hObject    handle to pushbuttonPolarX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

load_polarX;

assignin('base', 'TRUE_MEAN', TRUE_MEAN)
set(handles.editX, 'String', TRUE_MEAN(1));
set(handles.editY, 'String', TRUE_MEAN(2));
set(handles.editZ, 'String', TRUE_MEAN(3));

function editX_Callback(hObject, eventdata, handles)
% hObject    handle to editX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editX as text
%        str2double(get(hObject,'String')) returns contents of editX as a double
global TRUE_MEAN;
TRUE_MEAN(1) = str2double(get(hObject, 'String'));
if isnan(TRUE_MEAN(1))
    TRUE_MEAN(1) = 0;
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
assignin('base', 'TRUE_MEAN', TRUE_MEAN)

% --- Executes during object creation, after setting all properties.
function editX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function editY_Callback(hObject, eventdata, handles)
% hObject    handle to editY (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editY as text
%        str2double(get(hObject,'String')) returns contents of editY as a double

global TRUE_MEAN;

```

```

TRUE_MEAN(2) = str2double(get(hObject, 'String'));
if isnan(TRUE_MEAN(2))
    TRUE_MEAN(2) = 0;
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
assignin('base', 'TRUE_MEAN', TRUE_MEAN)

% --- Executes during object creation, after setting all properties.
function editY_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editY (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function editZ_Callback(hObject, eventdata, handles)
% hObject    handle to editZ (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of editZ as text
%        str2double(get(hObject,'String')) returns contents of editZ as a double
global TRUE_MEAN;
TRUE_MEAN(3) = str2double(get(hObject, 'String'));
if isnan(TRUE_MEAN(3))
    TRUE_MEAN(3) = 0;
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
assignin('base', 'TRUE_MEAN', TRUE_MEAN)

% --- Executes during object creation, after setting all properties.
function editZ_CreateFcn(hObject, eventdata, handles)
% hObject    handle to editZ (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in checkboxPt.
function checkboxPt_Callback(hObject, eventdata, handles)
% hObject    handle to checkboxPt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkboxPt

% --- Executes on button press in checkboxGt.
function checkboxGt_Callback(hObject, eventdata, handles)
% hObject    handle to checkboxGt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkboxGt

% --- Executes on button press in checkboxLp.
function checkboxLp_Callback(hObject, eventdata, handles)
% hObject    handle to checkboxLp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkboxLp

% --- Executes on button press in checkboxLa.

```

```

function checkboxLa_Callback(hObject, eventdata, handles)
% hObject      handle to checkboxLa (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkboxLa

% --- Executes on button press in checkboxWeighting.
function checkboxWeighting_Callback(hObject, eventdata, handles)
% hObject      handle to checkboxWeighting (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkboxWeighting
% --- Executes on button press in pushbuttonTestbed.
function pushbuttonTestbed_Callback(hObject, eventdata, handles)
% hObject      handle to pushbuttonTestbed (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

clear testbed_packets;
clear testbed;
global testbed_packets;
global testbed;
load testbed;
assignin('base', 'testbed_packets', testbed_packets)
assignin('base', 'testbed', testbed)

% --- Executes on button press in pushbuttonOK.
function pushbuttonOK_Callback(hObject, eventdata, handles)
% hObject      handle to pushbuttonOK (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global TRUE_MEAN;
global testbed_packets;
global testbed;
if get(handles.checkboxGt,'Value') == 1
    Include_SV_Radiation_Pattern_Compensation = 1;
end
if get(handles.checkboxLp,'Value') == 1
    Include_Space_Loss_Compensation = 1;
end
if get(handles.checkboxPt,'Value') == 1
    Include_SV_Power_Compensation = 1;
end
if get(handles.checkboxLa,'Value') == 1
    Include_Atm_Loss_Compensation = 1;
end
if get(handles.checkboxWeighting,'Value') == 1
    Include_Weight_Matrix = 1;
end

static_attitude;

% --- Executes on button press in pushbuttonCancel.
function pushbuttonCancel_Callback(hObject, eventdata, handles)
% hObject      handle to pushbuttonCancel (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close(gcf);

```